Bonus:

Sistema de memoria

Fundamentos de computadores II

José Manuel Mendías Cuadros
Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid

Contenidos

- ✓ Introducción.
- ✓ Principio de localidad.
- ✓ Jerarquía de memoria.
- ✓ Fundamentos de memoria cache.
- ✓ Memoria cache de emplazamiento directo.

- ✓ Apéndice tecnológico.

Transparencias basadas en los libros:

- S.L. Harris and D. Harris. *Digital Design and Computer Architecture. RISC-V Edition.*
- D.A. Patterson and J.L. Hennessy. *Computer Organization and Design. RISC-V Edition.*

Introducción

- Todo programador desea disponer de una memoria lo más rápida posible, del máximo tamaño y al mínimo precio.
- Cada tecnología tiene sus características, pero todas cumplen que:
  - A mayor capacidad, mayor tiempo de acceso.
  - A menor tiempo de acceso, mayor coste por byte.

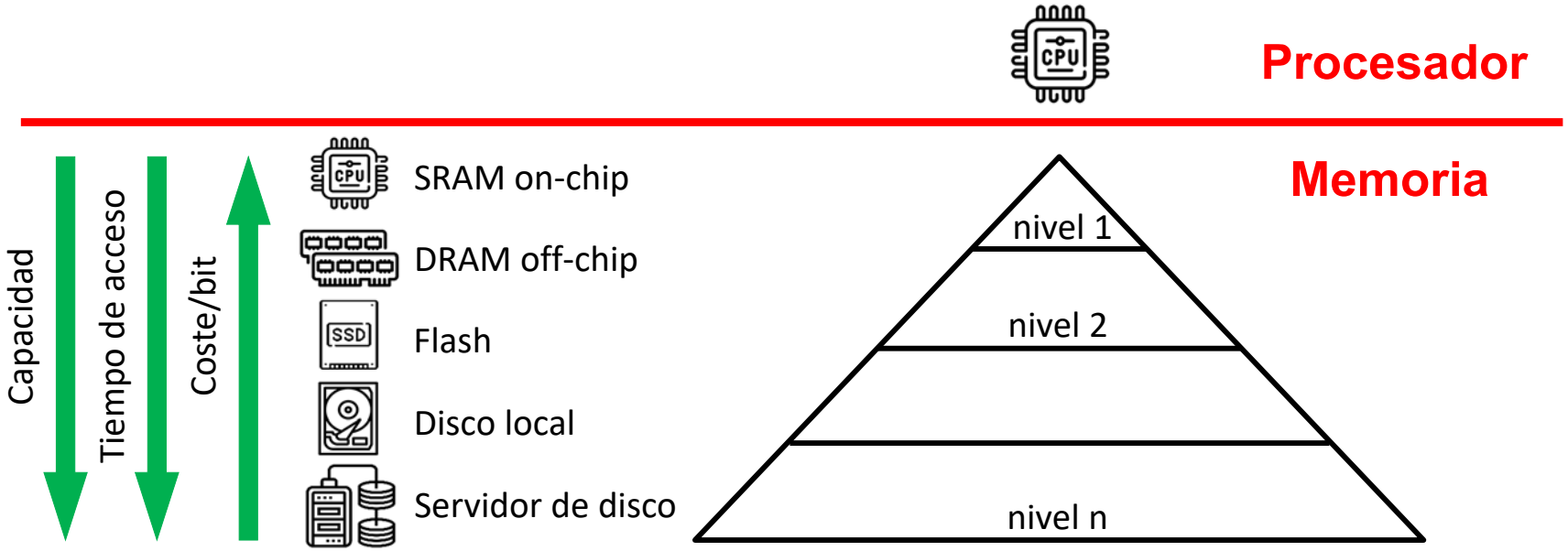
Tecnología	Capacidad típica	Tiempo de acceso (ns)	Coste ($/GiB)
SRAM semiconductora	10 KiB-10 MiB	0.5–2.5	500–1000
DRAM semiconductora	10 GiB	50–70	3–6
Flash semiconductora (Disco de estado sólido)	100 GiB	$5 \cdot 10^3$–$5 \cdot 10^4$	0.6–0.12
Disco magnético	10 TiB	$5 \cdot 10^6$–$2 \cdot 10^7$	0.01–0.02

fuente (adaptación): D.A. Patterson and J.L. Hennessy. Computer Organization and Design. RISC-V Edition (2021)

- Conclusión: es imposible satisfacer al programador con una única tecnología de memoria.

Introducción

- Un sistema de memoria moderno combina de manera transparente varias tecnologías de memoria organizadas por niveles jerárquicos.



- Desde el punto de vista del procesador esta estructura, con los mecanismos de gestión apropiados, se comporta como una única memoria con:
  - Tiempo de acceso medio cercano al de la tecnología más rápida (nivel 1).
  - Capacidad equivalente a la disponible en la tecnología de mayor capacidad (nivel n).
  - Coste/bit cercano al de la tecnología más barata (nivel n).

Principio de localidad

Introducción (i)

- Todo estudiante aprende su carrera leyendo libros.

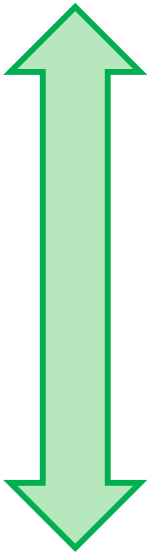
- Los libros de todas las carreras están en la biblioteca:
  - Para que los estudiantes puedan aprender cualquier disciplina.

Principio de localidad

Introducción (i)

- Todo estudiante aprende su carrera leyendo libros.

Si debe ir a la biblioteca a cambiar de libro cada vez que cambia de asignatura, derrocharía mucho tiempo.

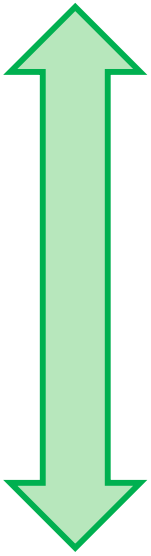


- Los libros de todas las carreras están en la biblioteca:
  - Para que los estudiantes puedan aprender cualquier disciplina.

Principio de localidad

Introducción (i)

- Todo estudiante aprende su carrera leyendo libros.



Si debe ir a la biblioteca a cambiar de libro cada vez que cambia de asignatura, derrocharía mucho tiempo.

La biblioteca tiene una enorme capacidad de almacenar libros, pero también es muy alto el tiempo de acceso a ellos.

- Los libros de todas las carreras están en la biblioteca:
  - Para que los estudiantes puedan aprender cualquier disciplina.

Principio de localidad

Introducción (ii)

- Para reducir este tiempo, acerca los libros que más usa.

Principio de localidad

Introducción (ii)

- Para reducir este tiempo, acerca los libros que más usa.

- Coloca libros del curso sobre la mesa de estudio:
  - Para tenerlos cerca cuando cambia de asignatura.
  - Caben pocos libros, pero es muy rápido acceder a ellos.

Principio de localidad

Introducción (ii)

- Para reducir este tiempo, acerca los libros que más usa.

- Coloca libros del curso sobre la mesa de estudio:
  - Para tenerlos cerca cuando cambia de asignatura.
  - Caben pocos libros, pero es muy rápido acceder a ellos.

- Coloca los libros del resto de la carrera en una librería:
  - Ordenados por curso por si hay algo que repasar.
  - Su capacidad es intermedia y el tiempo de acceso a la librería es mayor que al escritorio pero menor a la biblioteca.

- Solo irá a la biblioteca cuando necesite un libro distinto.

Principio de localidad

Introducción (iii)

- Esta solución reduce el tiempo medio de acceso a los libros.
  - Los que se usarán con mayor probabilidad están más cerca.
- Funciona porque los libros leídos no son aleatorios:
  - En cada momento, solo necesita un pequeño conjunto de los libros de la biblioteca.
  - Típicamente alterna la lectura de libros del mismo curso.
  - Ocasionalmente se leen libros de cursos pasados o de temáticas afines.
- Los libros leídos por un estudiante tienen:
  - Localidad temporal: si hoy lee un libro, es muy probable que en próximos días lea el mismo.
  - Localidad espacial: Si hoy lee un libro, es muy probable que en próximos días lea uno situado cerca.
    - Ya que los libros de la misma temática suelen colocarse juntos.

Principio de localidad

Introducción (iv)

- El principio de localidad es universal y se usa en infinidad de campos de la informática para mejorar el rendimiento, lo aplican:
  - Los programadores en ensamblador al decidir las variables que almacena en los registros del procesador, evitando el retraso del acceso a memoria.
  - Los navegadores Web al conservar copias locales de las páginas recientemente visitadas, evitando el retraso de la descarga.
  - Los servidores de Bases de Datos al conservar copias del resultado de las consultas más recurrentes, evitando el retraso de la consulta.
  - Los usuarios finales al decidir las aplicaciones que instala, al organizar sus documentos, al hacer backups de sus archivos, etc..

- En particular el principio de localidad se usa para gestionar la jerarquía de memoria de un computador moderno.

Principio de localidad

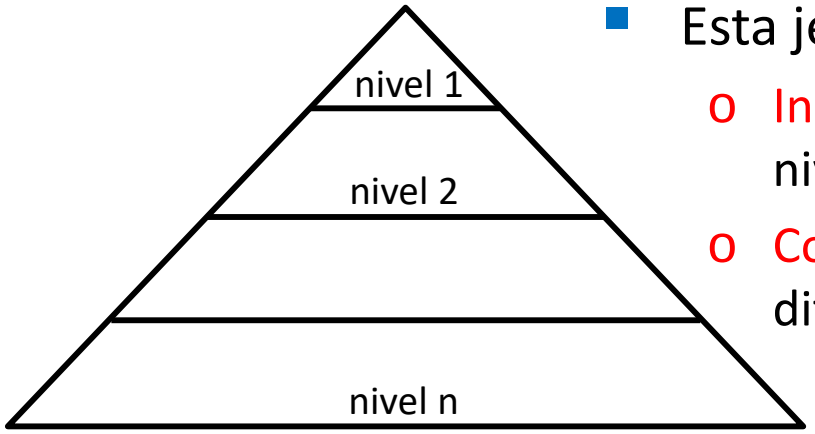
Localidad de referencias

- En todo intervalo de tiempo, un procesador solo usa un pequeño conjunto de los datos/instrucciones del programa.
- Las referencias a memoria que hace no son aleatorias ya que tienen:
  - Localidad temporal: si direcciona un dato/instrucción es muy probable que en el futuro próximo direccione el mismo.
    - Los programas tienen bucles que ejecutan varias veces las mismas instrucciones.
    - Los programas se estructuran en funciones que se ejecutan recurrentemente.
    - Las variables almacenadas en memoria son leídas/escritas recurrentemente.
  - Localidad espacial: si direcciona un dato/instrucción es muy probable que en el futuro próximo direccione los ubicados en direcciones próximas.
    - Las instrucciones de un programa se ubican en direcciones consecutivas y se ejecutan normalmente en secuencia.
    - Las variables locales a una función se ubican en direcciones consecutivas del marco de pila y se alterna la lectura/escritura de todas ellas.
    - Los arrays se ubican en direcciones consecutivas y, además es común procesar sus elementos secuencialmente.

Jerarquía de memoria

Principios de funcionamiento

- El objetivo de un sistema jerárquico de memoria es:
  - Asegurar que las referencias (datos/instrucciones) que en todo momento necesita el procesador estén en el nivel más alto de jerarquía.
- Lo consigue haciendo uso del principio de localidad:
  - Manteniendo en cada nivel una copia de un subconjunto de las referencias almacenadas en el nivel inmediatamente inferior.
  - Manteniendo cerca del procesador las últimas referencias solicitadas.
  - Moviendo cerca del procesador no sólo la referencia solicitada sino también sus contiguas.



- Esta jerarquía tiene 2 propiedades:
  - Inclusión: todos las referencias contenidas en un nivel encuentran en todos los niveles inferiores.
  - Coherencia: las copias de una misma referencia en diferentes niveles deben ser coherentes.

Jerarquía de memoria

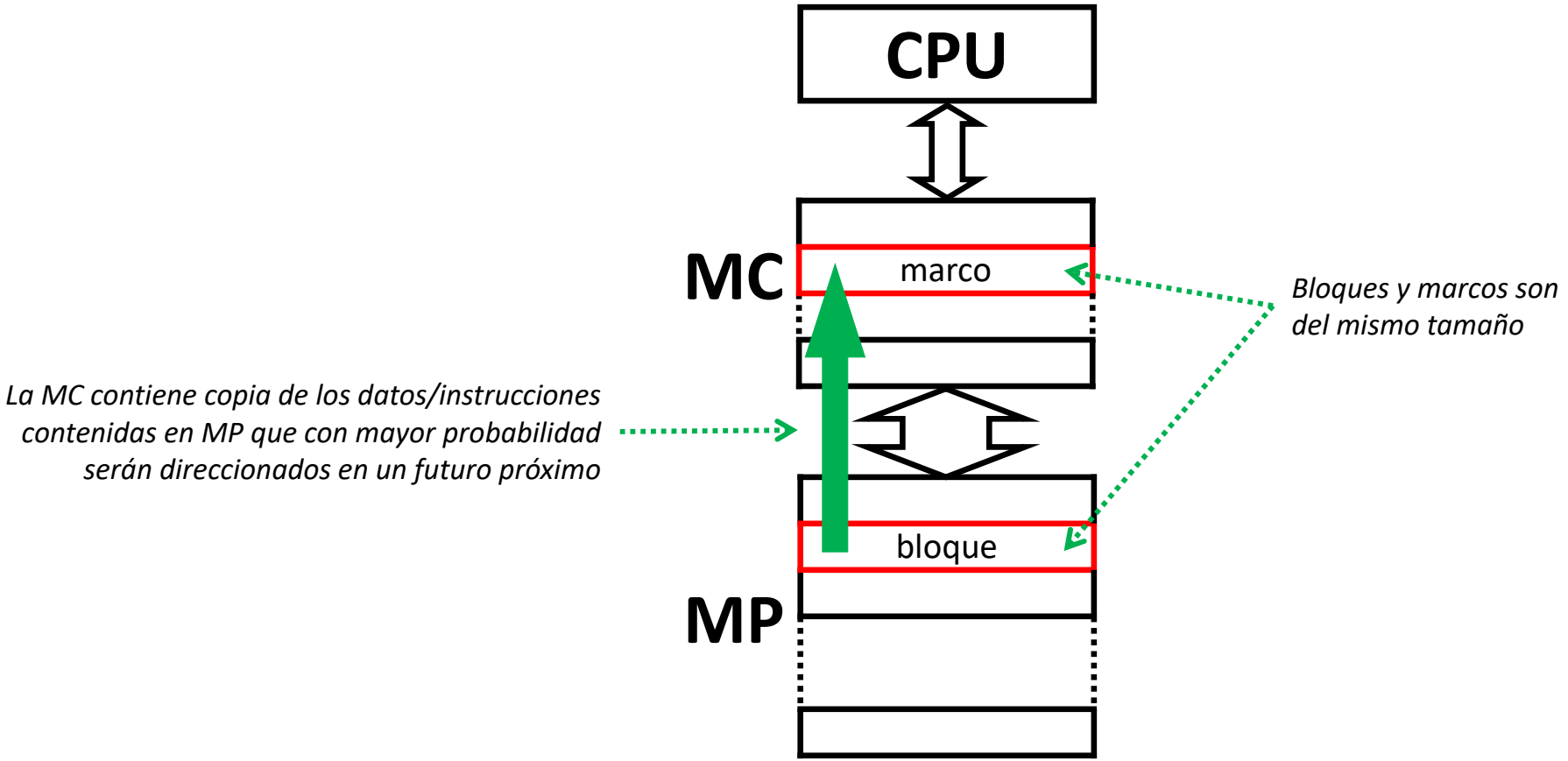
Ejemplo

- La jerarquía de memoria de un computador típico tiene muchos niveles
  - Que pueden ser más si los discos incorporan sus propias caches para acelerar el tiempo medio de acceso a sus datos.

Nivel	Ubicación	Tecnología	Tiempo de acceso	Capacidad	Gestionado por
Registros	Ruta de datos	flip-flop CMOS	1 ciclo	1 KiB	Compilador
Cache nivel 1	On-chip	SRAM	2-4 ciclos	64 KiB	Hardware
Cache nivel 2	On-chip	SRAM	10 ciclos	256 KiB	Hardware
Cache nivel 3	On-chip	SRAM	40 ciclos	16 MiB	Hardware
Memoria principal	Off-chip	DRAM	200 ciclos	16 GiB	SO
Disco Flash	Periférico	Flash	~µs	512 GiB	SO
Disco duro	Periférico	Magnética	~ms	2 TiB	SO

Fundamentos de memoria cache

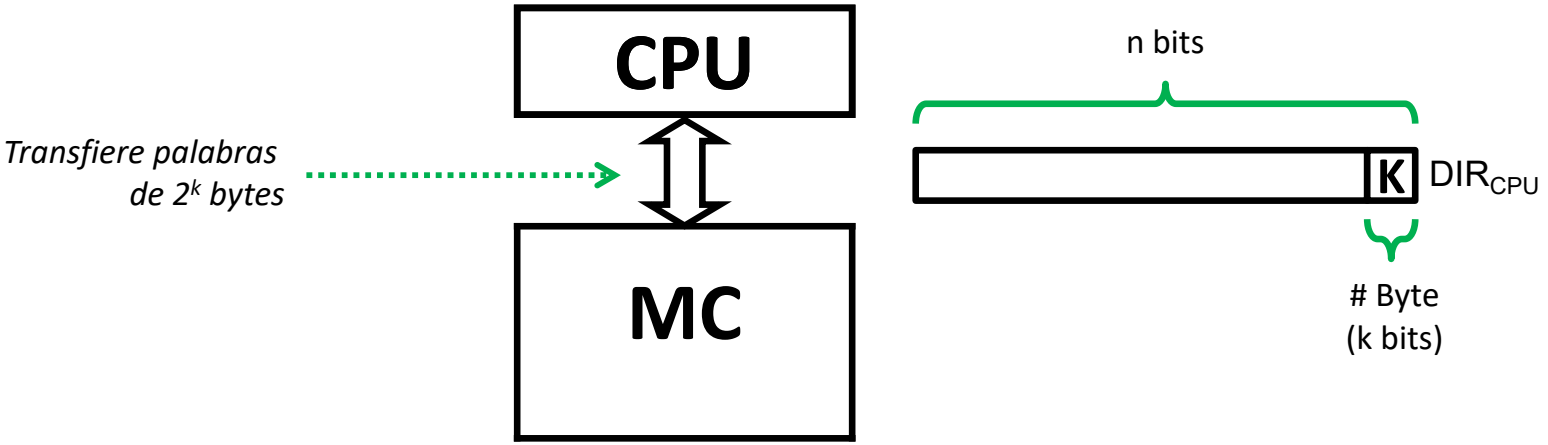
- Analicemos una jerarquía de memoria elemental de 2 niveles:
  - Formada por una memoria cache (MC) y una memoria principal (MP).
  - La MC es una memoria pequeña y rápida cuyo objetivo reducir el tiempo medio de acceso a los datos/instrucciones almacenados en MP.



Fundamentos de memoria cache

Organización

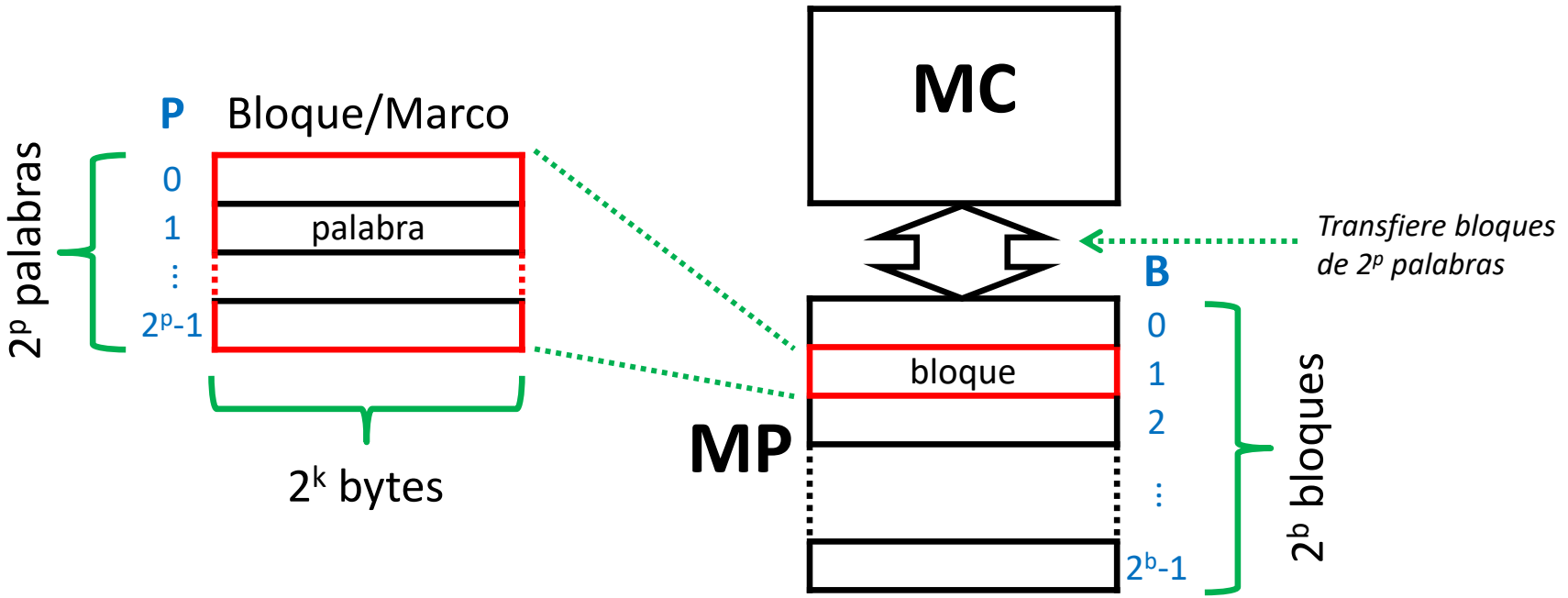
- Para el procesador la memoria cache es transparente:
  - El procesador envía direcciones de n bits.
  - Por cada dirección, la MC transfiere la palabra de 2^k bytes correspondiente.



Fundamentos de memoria cache

Organización de memoria principal (i)

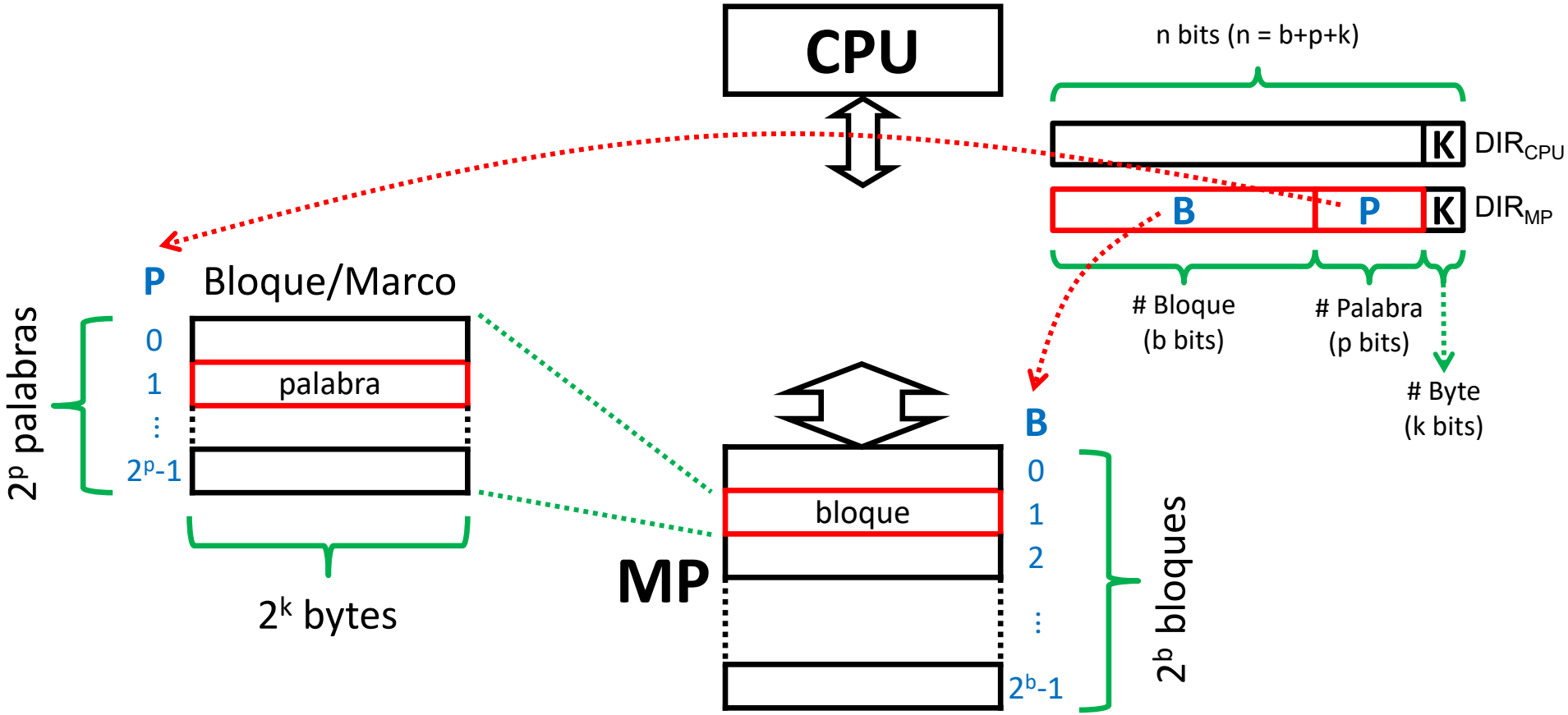
- La memoria principal (direccionable por bytes):
  - Tiene capacidad para 2^{n-k} palabras de 2^k bytes cada una (2^n bytes).
  - Está "dividida" en nB = 2^b bloques de tamaño fijo de 2^p palabras por bloque.
  - Por cada dirección, la MP transfiere a la MC un bloque al completo.
    - El bloque contiene la palabra direccionada y sus contiguas (localidad espacial).



Fundamentos de memoria cache

Organización de memoria principal (ii)

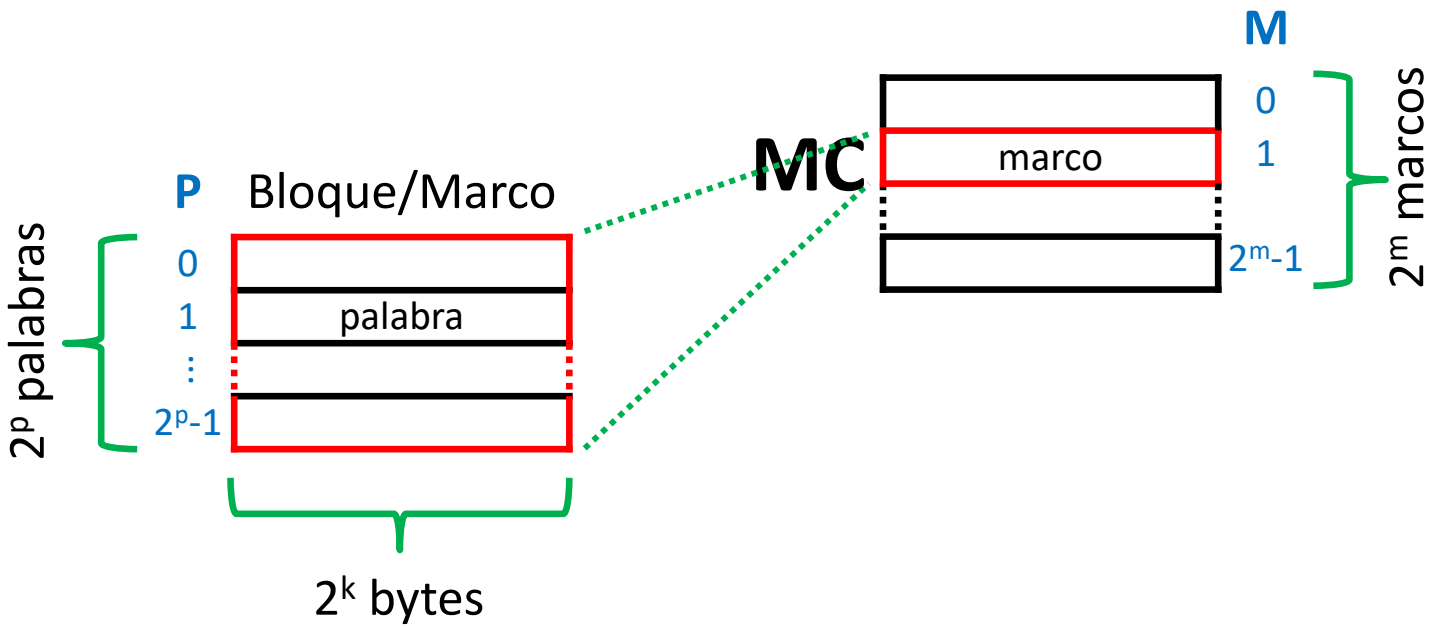
- Existe una relación inmediata entre la dirección de la palabra y:
  - El número de bloque de MP que la contiene y que es transferido a MC.
  - Su posición relativa dentro del bloque.



Fundamentos de memoria cache

Organización de memoria cache (i)

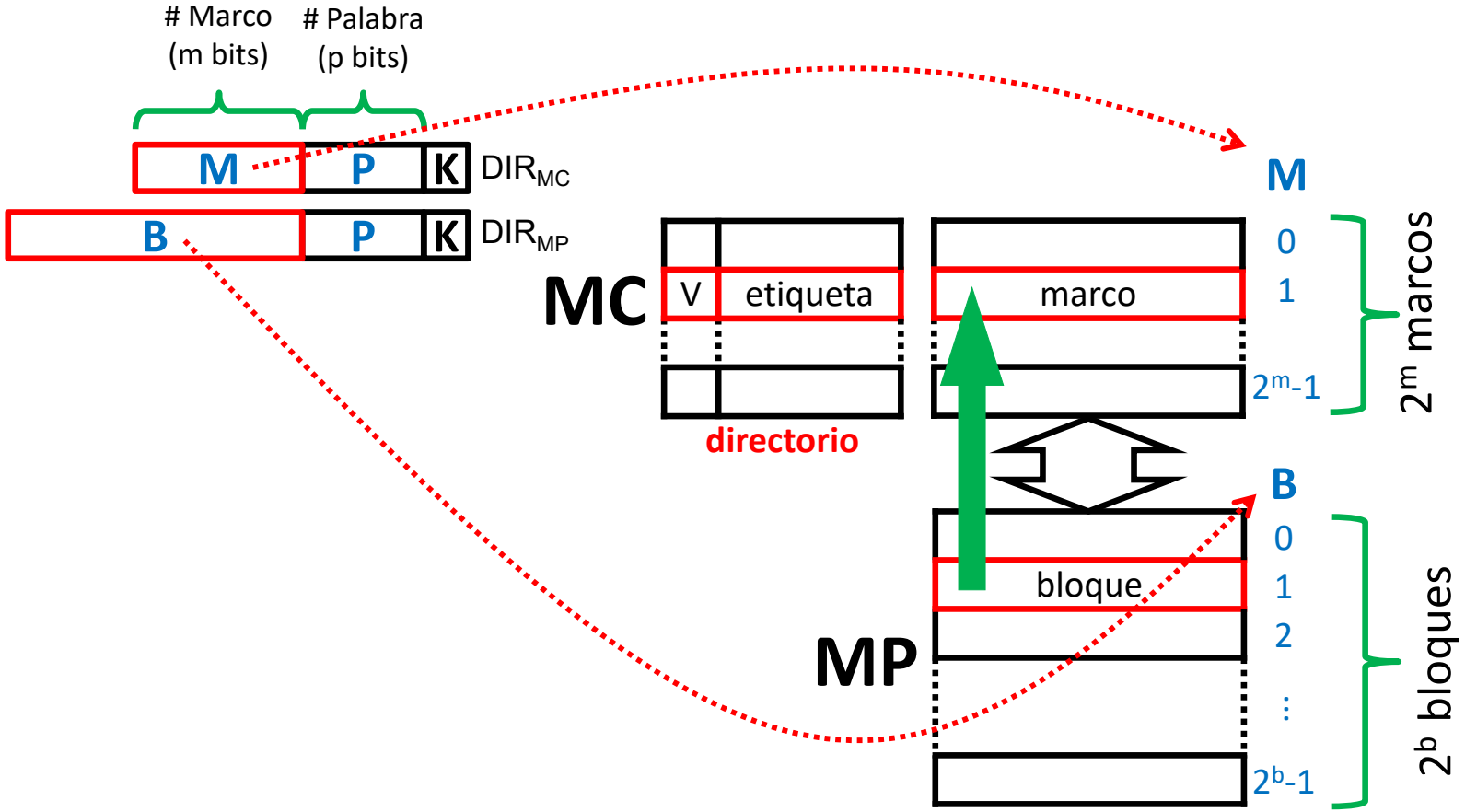
- La memoria cache:
  - Está "dividida" en nM = 2^m marcos de bloque (líneas de cache) de tamaño fijo de 2^p palabras por marco (nM << nB).
  - Tiene capacidad para 2^{m+p} palabras de 2^k bytes cada una (2^{m+p+k} bytes).
  - Cada marco almacena un bloque distinto de MP.
    - Y lo conserva hasta que sea reemplazado por otro (localidad temporal).



Fundamentos de memoria cache

Organización de memoria cache (ii)

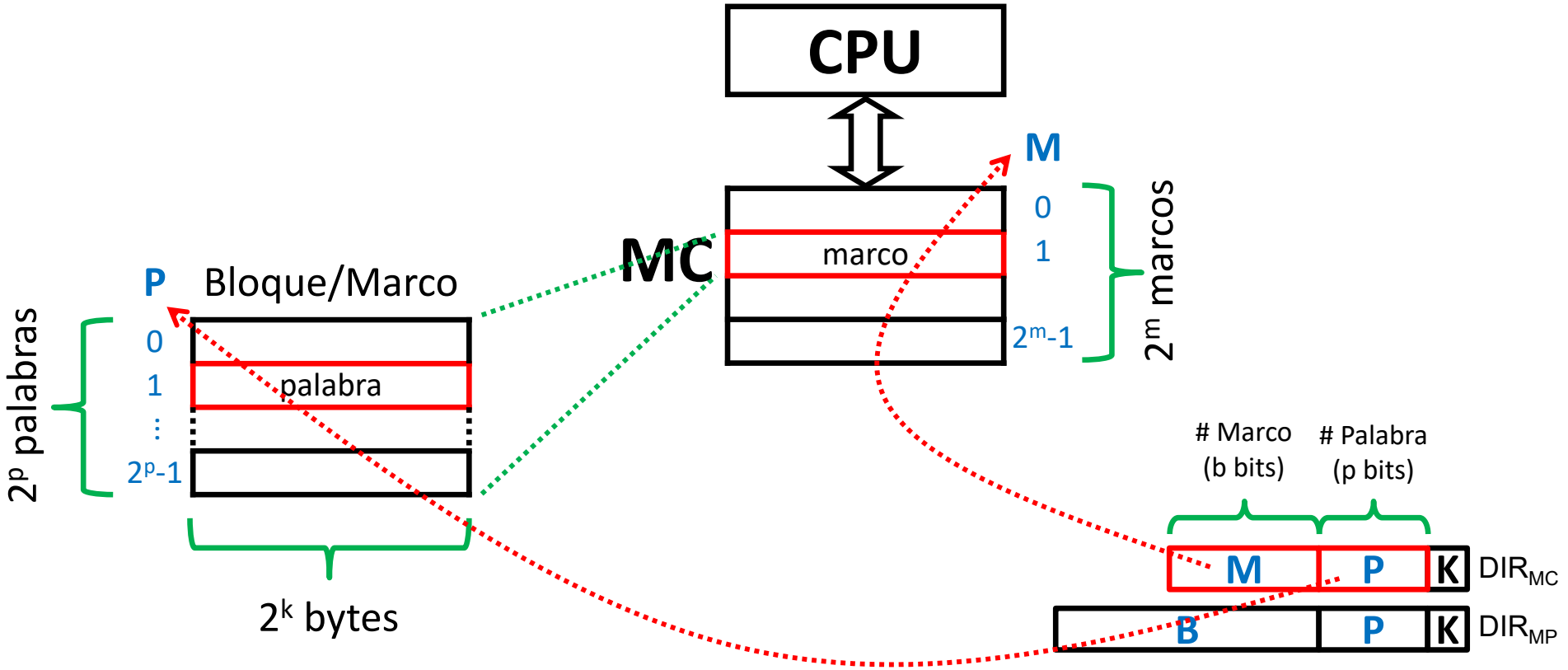
- Además, la MC dispone de un directorio en el que, cada marco, tiene:
  - Una etiqueta indicando el número de bloque de MP del que tiene copia.
  - Un bit de validez indicando si contiene o no un bloque válido.
    - Inicialmente vale 0, cuando el marco se rellena se pone a 1



Fundamentos de memoria cache

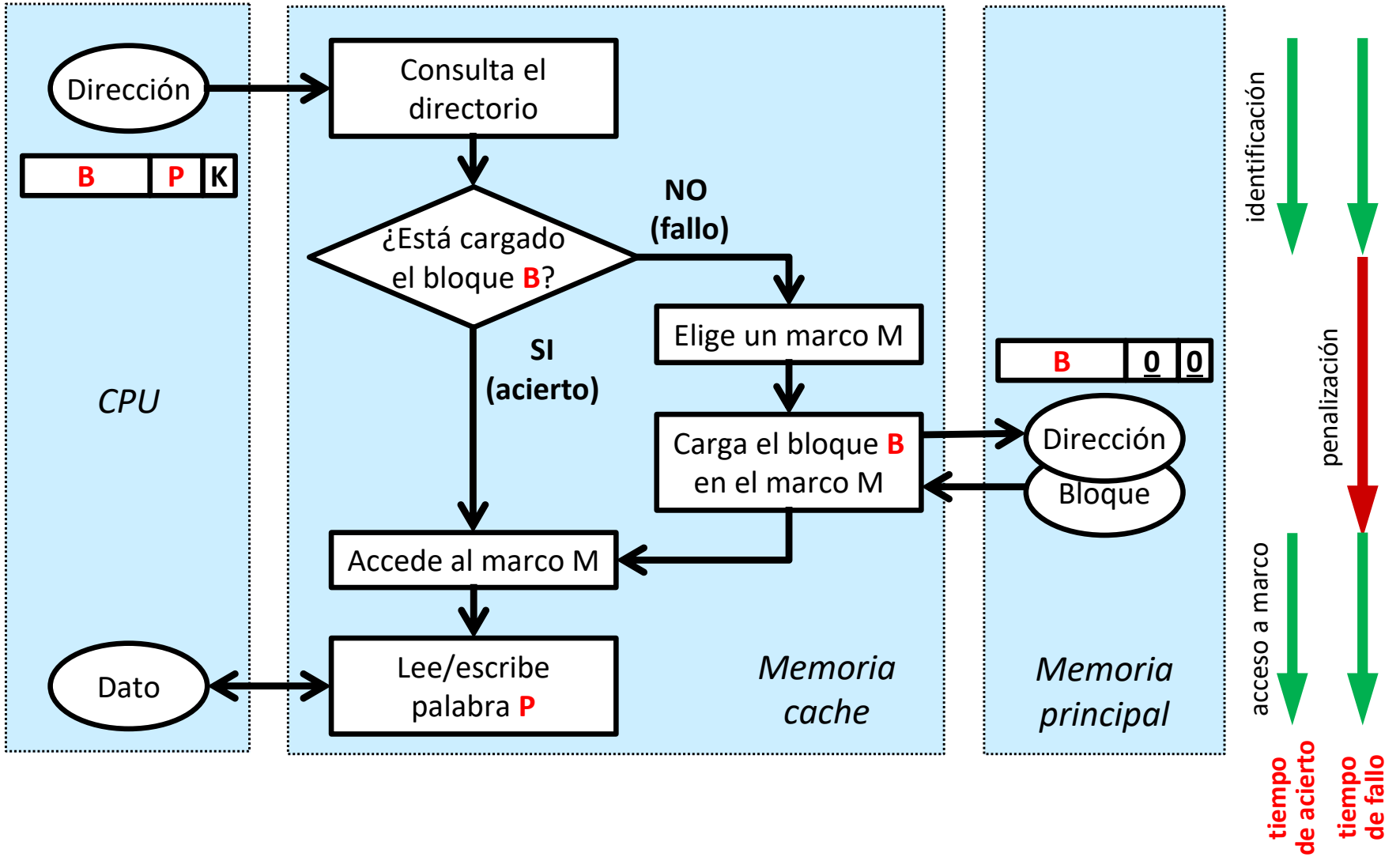
Organización de memoria cache (iii)

- Conocido el marco en donde se ubica cada bloque, la obtención de la posición relativa de la palabra dentro del marco es inmediata.



Fundamentos de memoria cache

Ciclo de acceso



Fundamentos de memoria cache

Caracterización (i)

- Toda referencia (dato/instrucción) generada por la CPU puede ser un:
  - Acierto (*hit*): la referencia solicitada está en MC.
    - Tasa de aciertos (*hit ratio, H*): fracción de referencias encontradas en MC, $H = \frac{n_h}{n}$
    - Tiempo de acierto (t_h): tiempo identificación + tiempo de acceso a MC.
  - Fallo (*miss*): la referencia solicitada no está en MC y hay que cargar de MP el bloque que lo contiene.
    - Tasa de fallos (*miss ratio, F*): $F = \frac{n_m}{n} = \frac{(n - n_h)}{n} = 1 - H$
    - Tiempo de fallo: tiempo de acierto + tiempo de penalización
    - Tiempo de penalización (t_p): tiempo de reemplazamiento de un bloque cargado en MC por el de MP que contiene la referencia solicitada.

- El tiempo medio de acceso a MC (T_{MAM}):

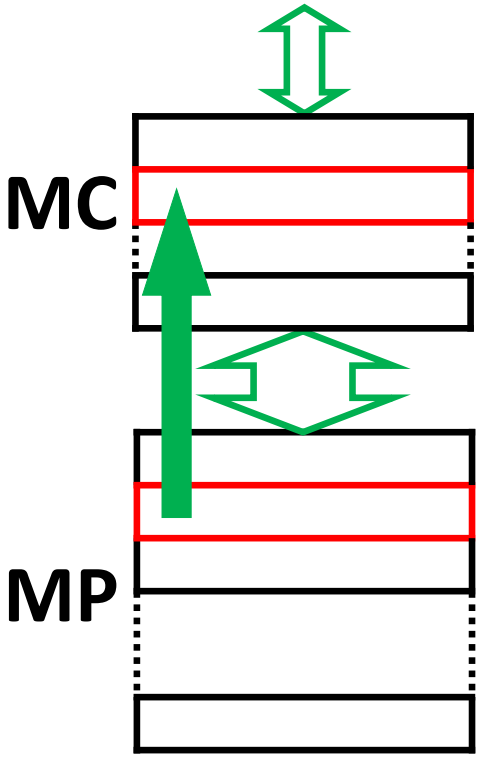
$$t_{MAM} = \frac{n_h}{n} \cdot t_h + \frac{n_m}{n} \cdot (t_h + t_p) = H \cdot t_h + (1 - H) \cdot (t_h + t_p)$$

$$t_{MAM} = t_h + (1 - H) t_p$$

Fundamentos de memoria cache

Caracterización (ii)

- MP y MC están implementadas con tecnologías diferentes
  - MP utiliza DRAM y MC utiliza SRAM.
  - Cada una con su Capacidad (S), Coste por byte (C) y Tiempo de acceso (T).
- El par MC+MP en conjunto estará caracterizado por:



Capacidad equivalente a la de la memoria de mayor capacidad (MP) por la propiedad de inclusión

$$S = S_{MP}$$

Coste/bit cercano al de la memoria con tecnología más barata (MP)

$$C = \frac{C_{SRAM} \cdot S_{MC} + C_{DRAM} \cdot S_{MP}}{S_{MC} + S_{MP}} \quad si\ S_{MP} \gg S_{MC} \Rightarrow C \approx C_{DRAM}$$

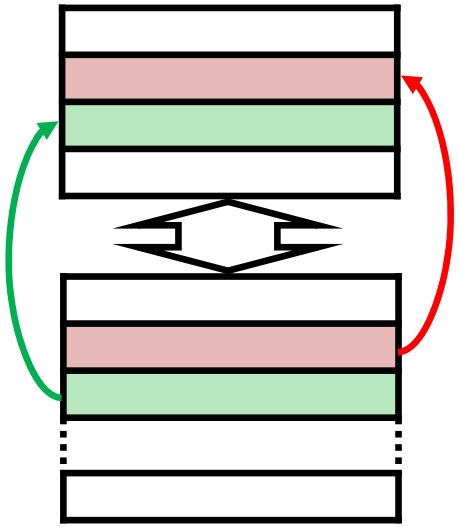
$$T = T_{MAM} = t_h + (1 - H)t_p \quad si \begin{Bmatrix} H \approx 1 \\ t_h \approx T_{SRAM} \end{Bmatrix} \Rightarrow T \approx T_{SRAM}$$

Tiempo de acceso cercano al de la memoria con tecnología más rápida (MC)

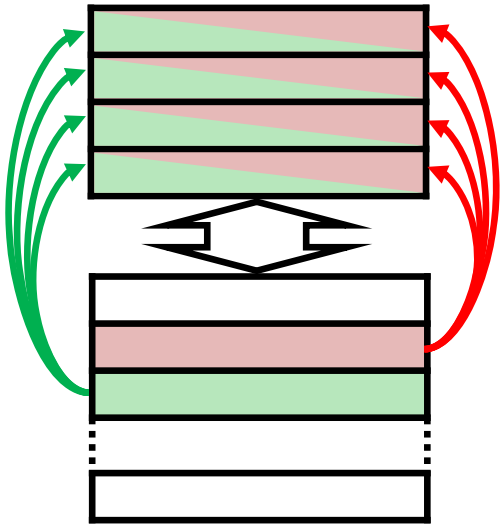
Fundamentos de memoria cache

Aspectos básicos de diseño (i)

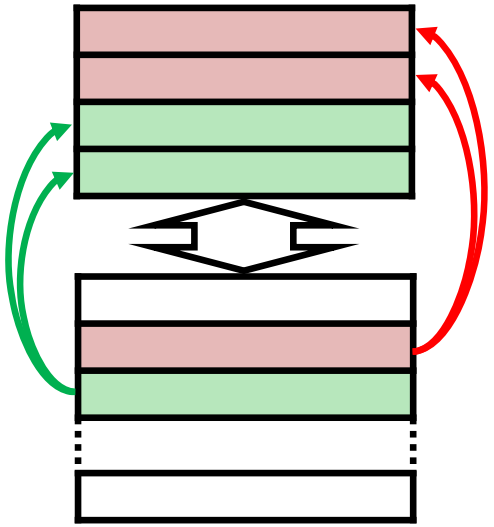
- Política de emplazamiento: determina en qué marcos de MC puede cargarse cada bloque de MP.
  - Necesaria porque existen menos marcos en MC que bloques en MP.
- Tipos:
  - Directo: cada bloque solo puede cargarse en un único marco.
  - Asociativo: cada bloque puede cargarse en cualquier marco.
  - Asociativo por conjuntos: cada bloque puede cargarse en cualquiera de los marcos de un conjunto determinado de marcos.



Directo



Asociativo



Asociativo por conjuntos

Fundamentos de memoria cache

Aspectos básicos de diseño (ii)

- Política de reemplazamiento: determina qué bloque cargado sustituir cuando están ocupados todos los marcos en donde puede cargarse.
  - Necesaria si un bloque puede cargarse en más de un marco.
  - No aplica a MC de emplazamiento directo.
- Tipos:
  - Aleatorio: reemplaza un bloque escogido al azar.
  - FIFO (*first in first out*): reemplaza el bloque que lleve más tiempo cargado.
  - LRU (*least recently used*): reemplaza el que lleve más tiempo sin referenciase.

Fundamentos de memoria cache

Aspectos básicos de diseño (iii)

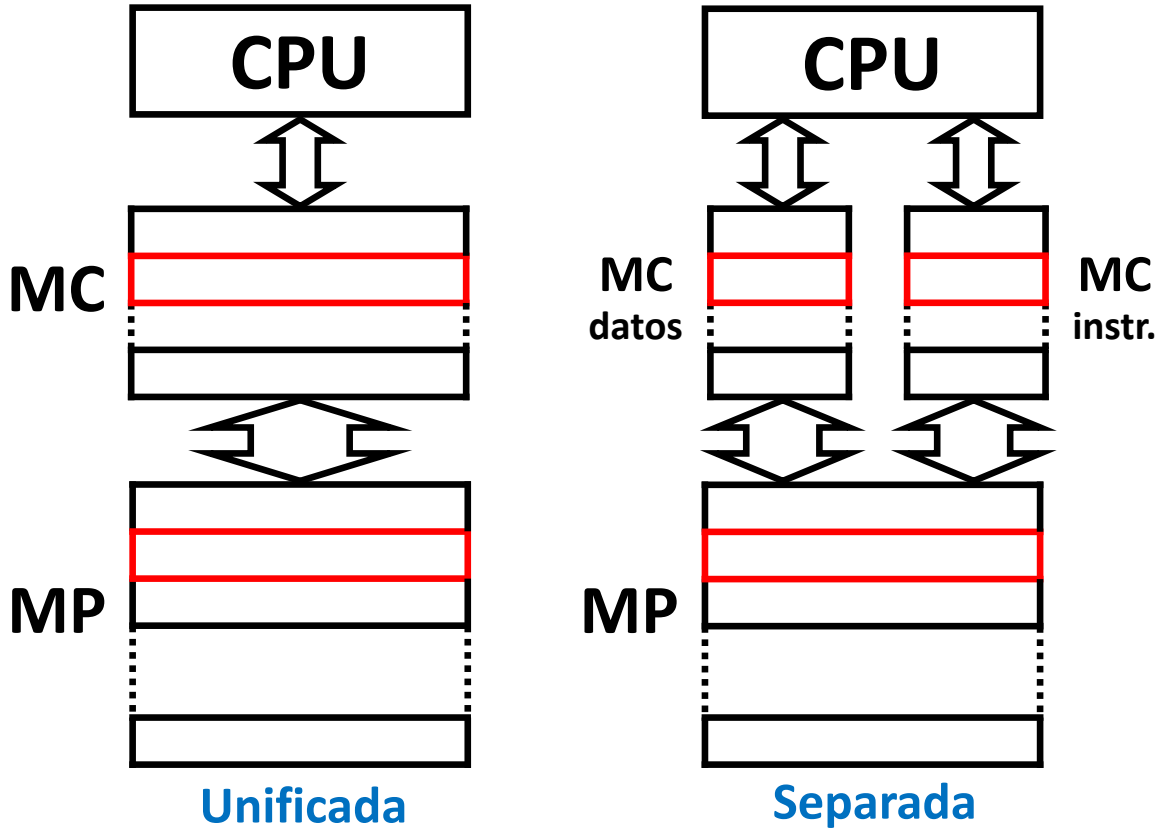
- Cuando la CPU escribe un dato, éste se almacena en la MC.
  - Si hay acierto, se escribe en el bloque cargado que corresponda.
  - Si hay fallo, se carga en MC el nuevo bloque y se realiza la escritura.

- Política de actualización: determina cuándo se actualiza en MP un bloque modificado en MC.
  - Necesaria para mantener la coherencia entre la MC y la MP.
- Tipos:
  - Escritura inmediata (*write-through*): la MP se actualiza a la vez que se escribe en el bloque cargado en MC.
  - Post-escritura (*write-back*): la MP sólo se actualiza cuando se reemplaza el bloque modificado.

Fundamentos de memoria cache

Aspectos básicos de diseño (iv)

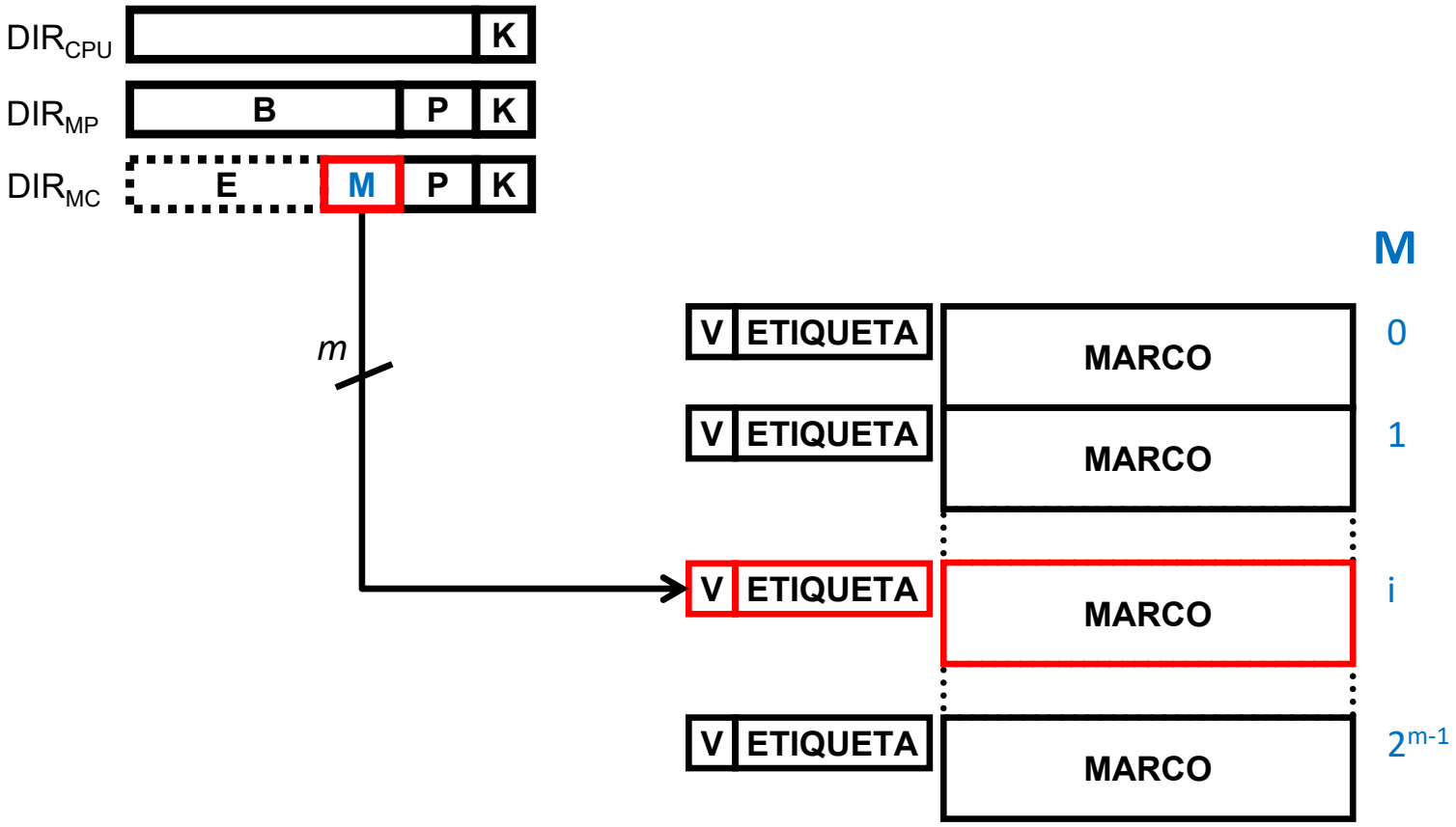
- Cache unificada: existe una única MC común para datos e instrucciones.
- Cache separada: existen dos MC una para datos y otra para instrucciones.
  - Cada una puede implementarse con políticas diferentes.



MC de emplazamiento directo

Organización (i)

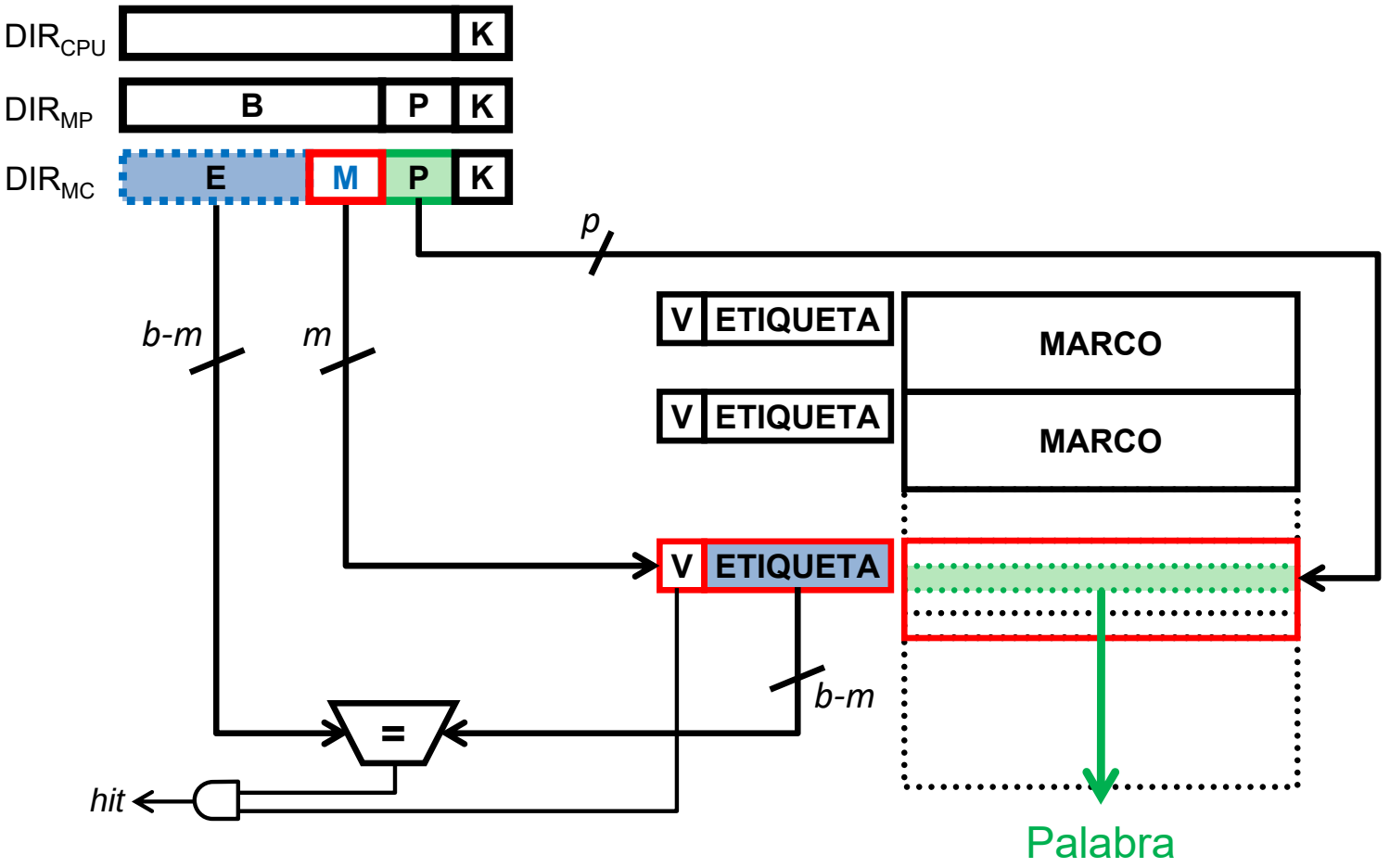
- Un bloque B se carga siempre en el marco M = B mod nM
  - Como nM = 2^m entonces M = "m bits menos significativos de B"
  - El acceso al marco y al directorio es directo y se realiza en paralelo.



MC de emplazamiento directo

Organización (ii)

- Para cada marco, el directorio almacena una etiqueta con los b-m bits restantes de la dirección del bloque almacenado.
  - Basta comparar las etiquetas para saber si un bloque está cargado en MC.

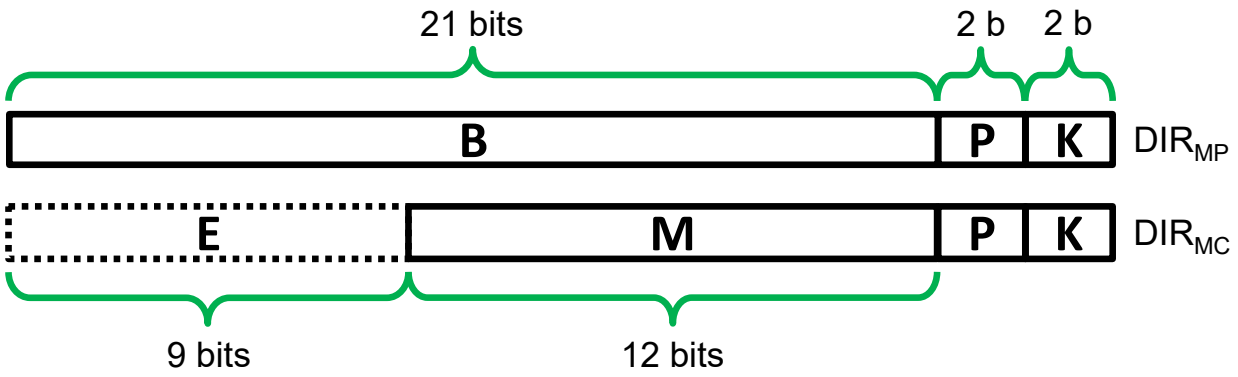


MC de emplazamiento directo

Ejemplo de dimensionamiento

- Memoria principal: 32 MB (2^{25}) direccionable por bytes
- Tamaño de palabra: 4 B
- Tamaño de bloque: 16 B (4 palabras)
- Memoria cache de emplazamiento directo: 64 KB

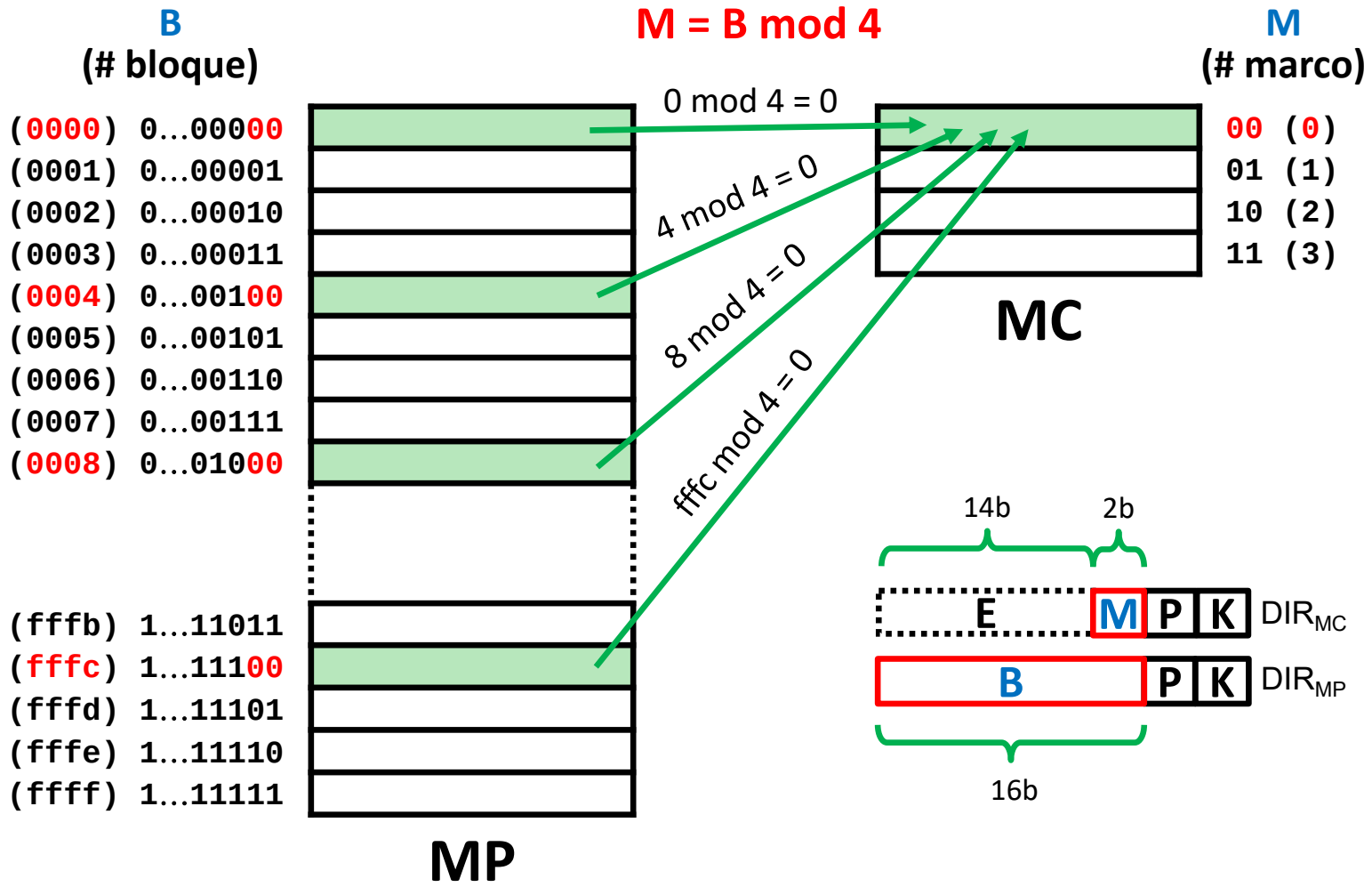
- Número de bloques en MP (nB) = 32 MB / 16 B = 2 M (2^{21})
- Número de marcos en MC (nM) = 64 KB / 16 B = 4 K (2^{12})
- Anchura de etiqueta = 21 – 12 = 9



MC de emplazamiento directo

Ejemplo de emplazamiento (i)

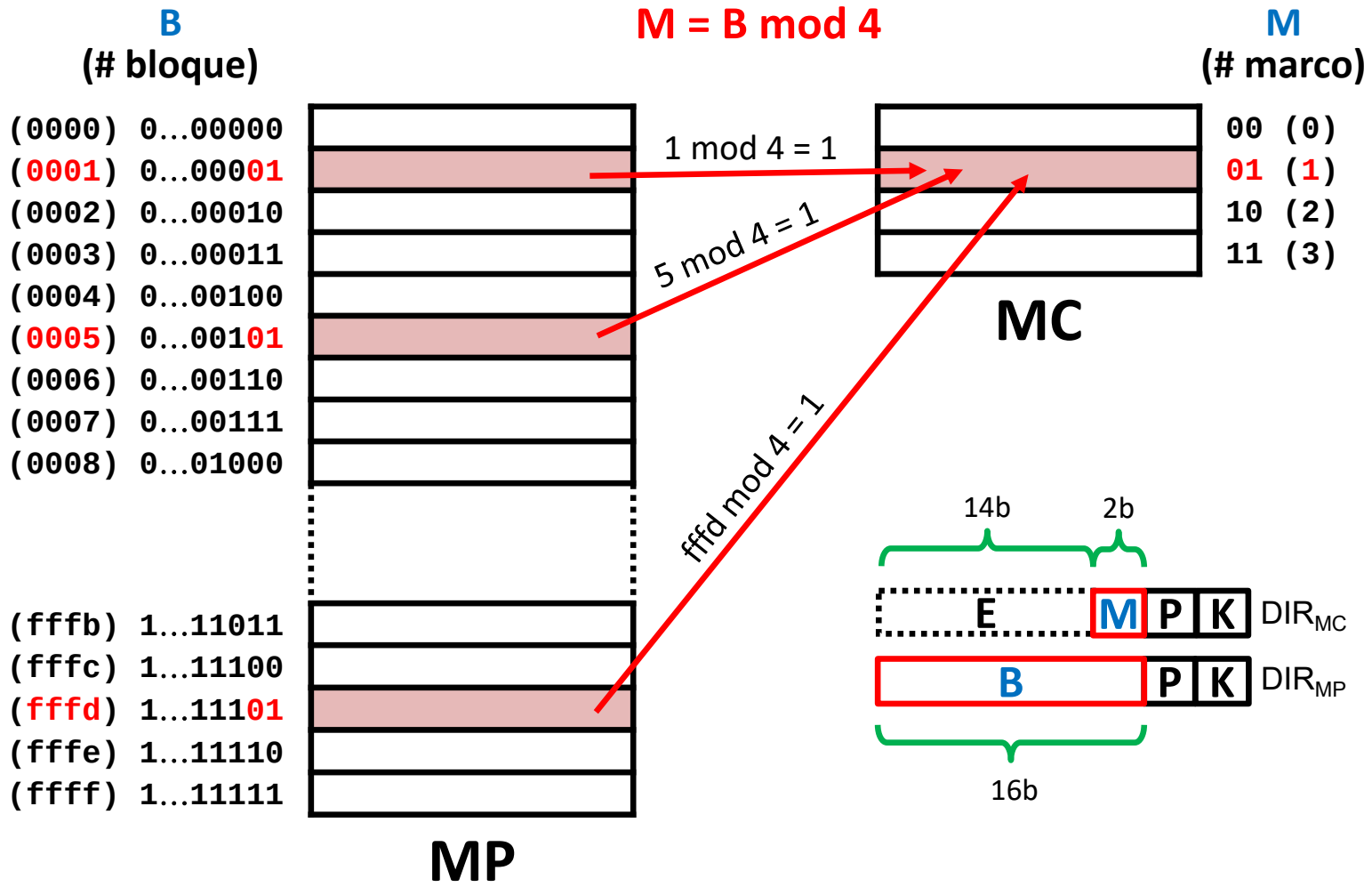
- Emplazamiento directo de una MP 2^{16} bloques en una MC de 4 marcos
  - $2^{16} \div 4 = 2^{14}$ bloques de MP distintos pueden cargarse en el mismo marco MC



MC de emplazamiento directo

Ejemplo de emplazamiento (ii)

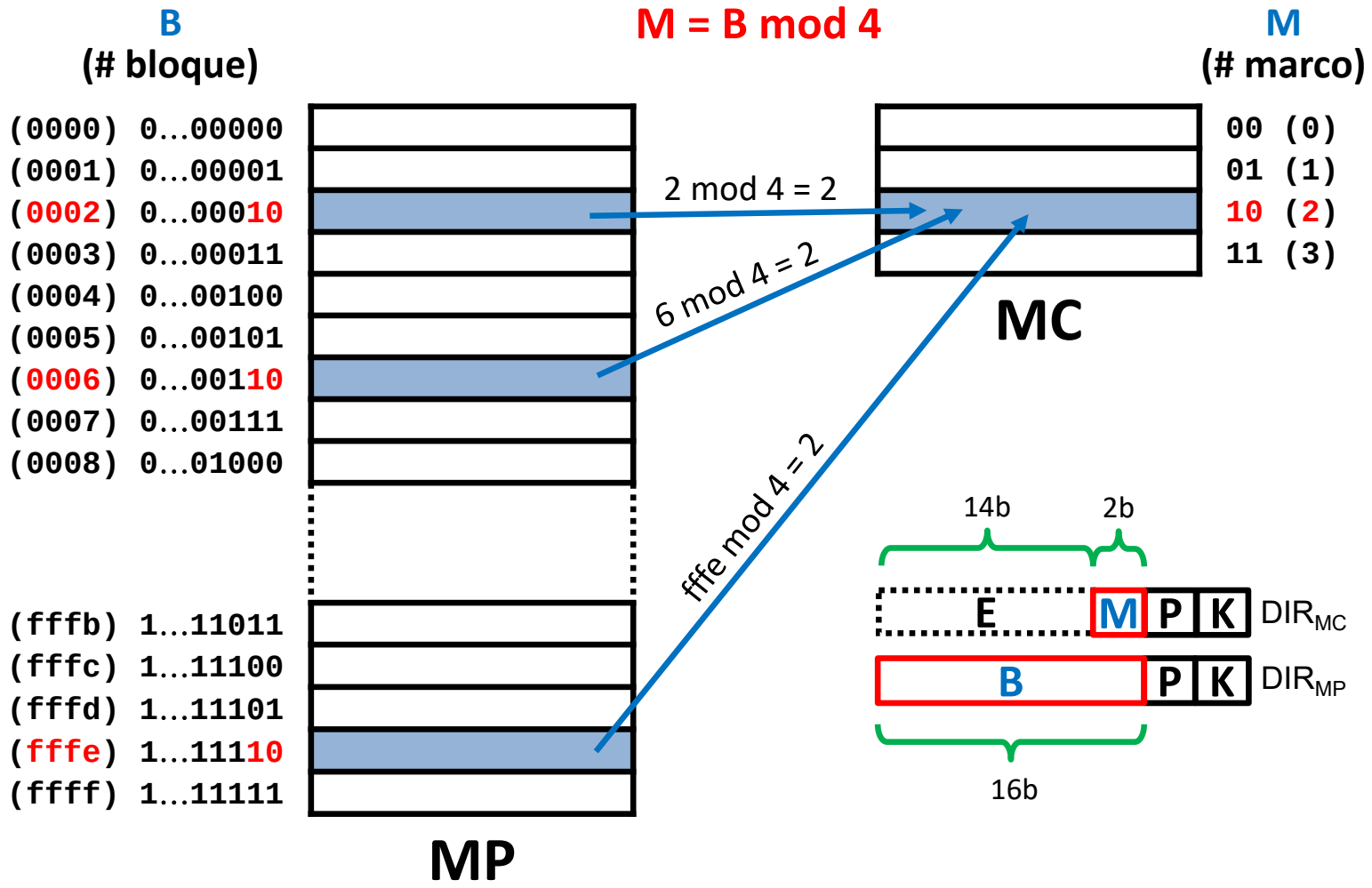
- Emplazamiento directo de una MP 2^{16} bloques en una MC de 4 marcos
  - $2^{16} \div 4 = 2^{14}$ bloques de MP distintos pueden cargarse en el mismo marco MC



MC de emplazamiento directo

Ejemplo de emplazamiento (iii)

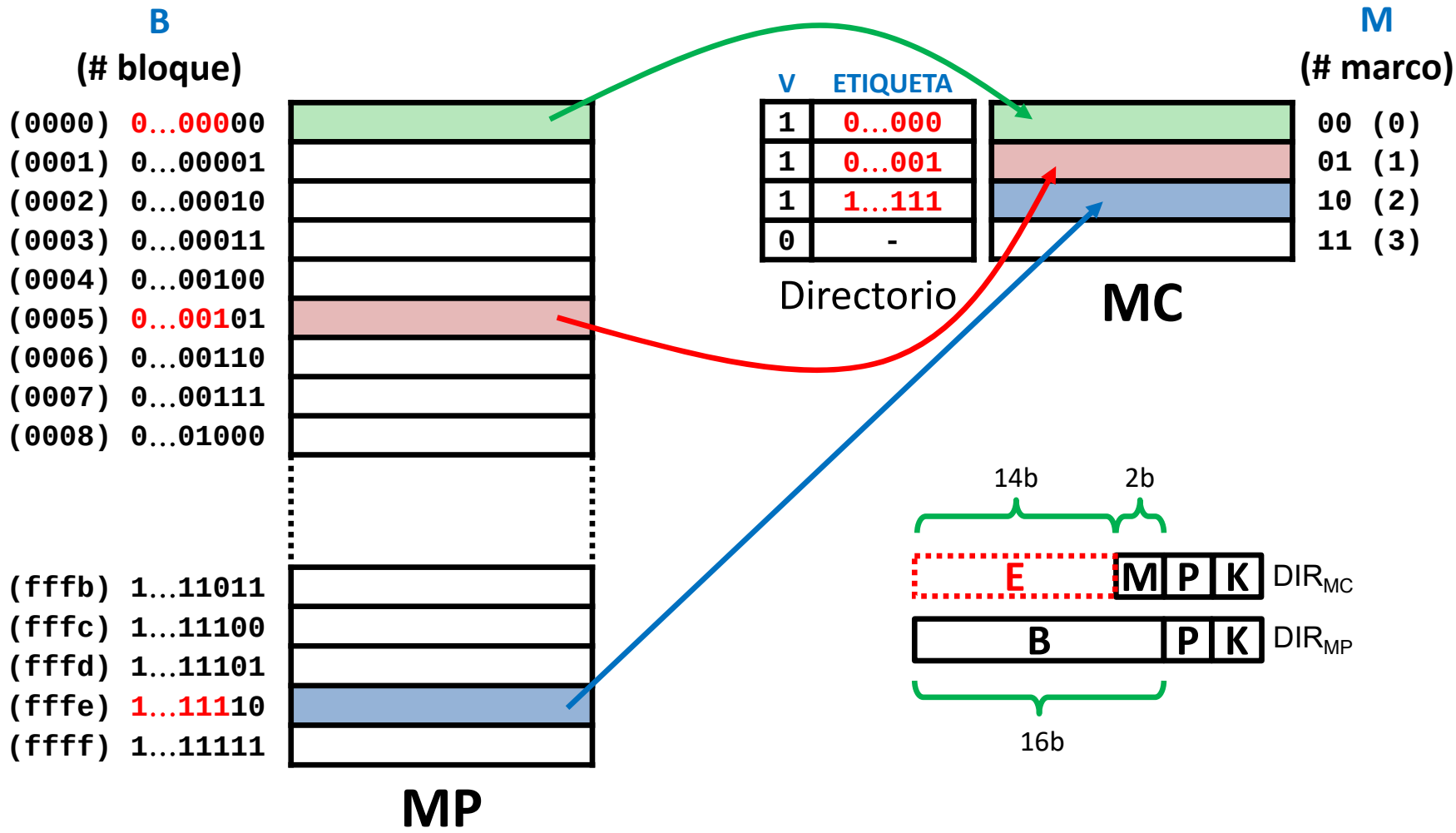
- Emplazamiento directo de una MP 2^{16} bloques en una MC de 4 marcos
  - $2^{16} \div 4 = 2^{14}$ bloques de MP distintos pueden cargarse en el mismo marco MC



MC de emplazamiento directo

Ejemplo de emplazamiento (iv)

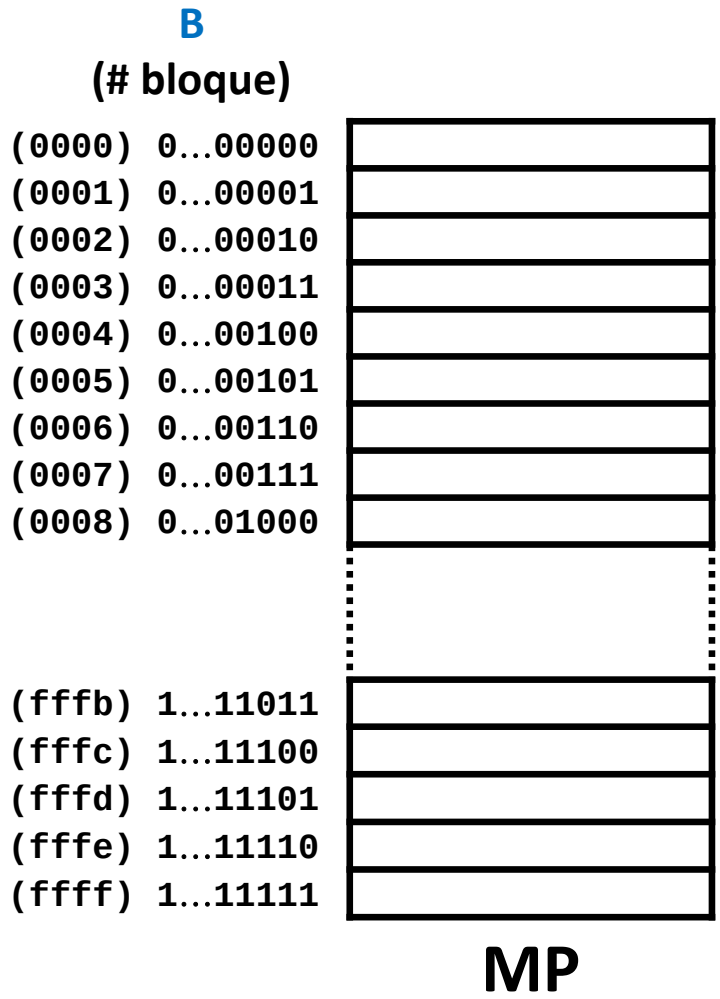
- Emplazamiento directo de una MP 2^{16} bloques en una MC de 4 marcos
  - En todo momento, un máximo de 4 bloques de MP están cargados en MC

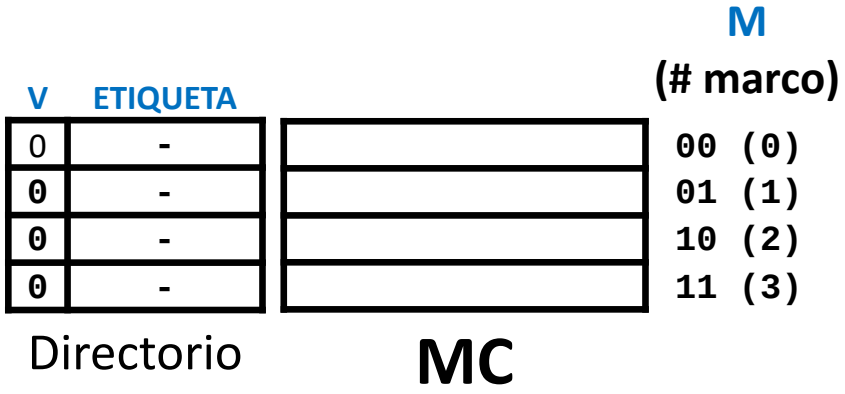


MC de emplazamiento directo

Ejemplo de fallos por conflicto (i)

- Una MC con emplazamiento directo puede dar fallo aunque esté prácticamente vacía.





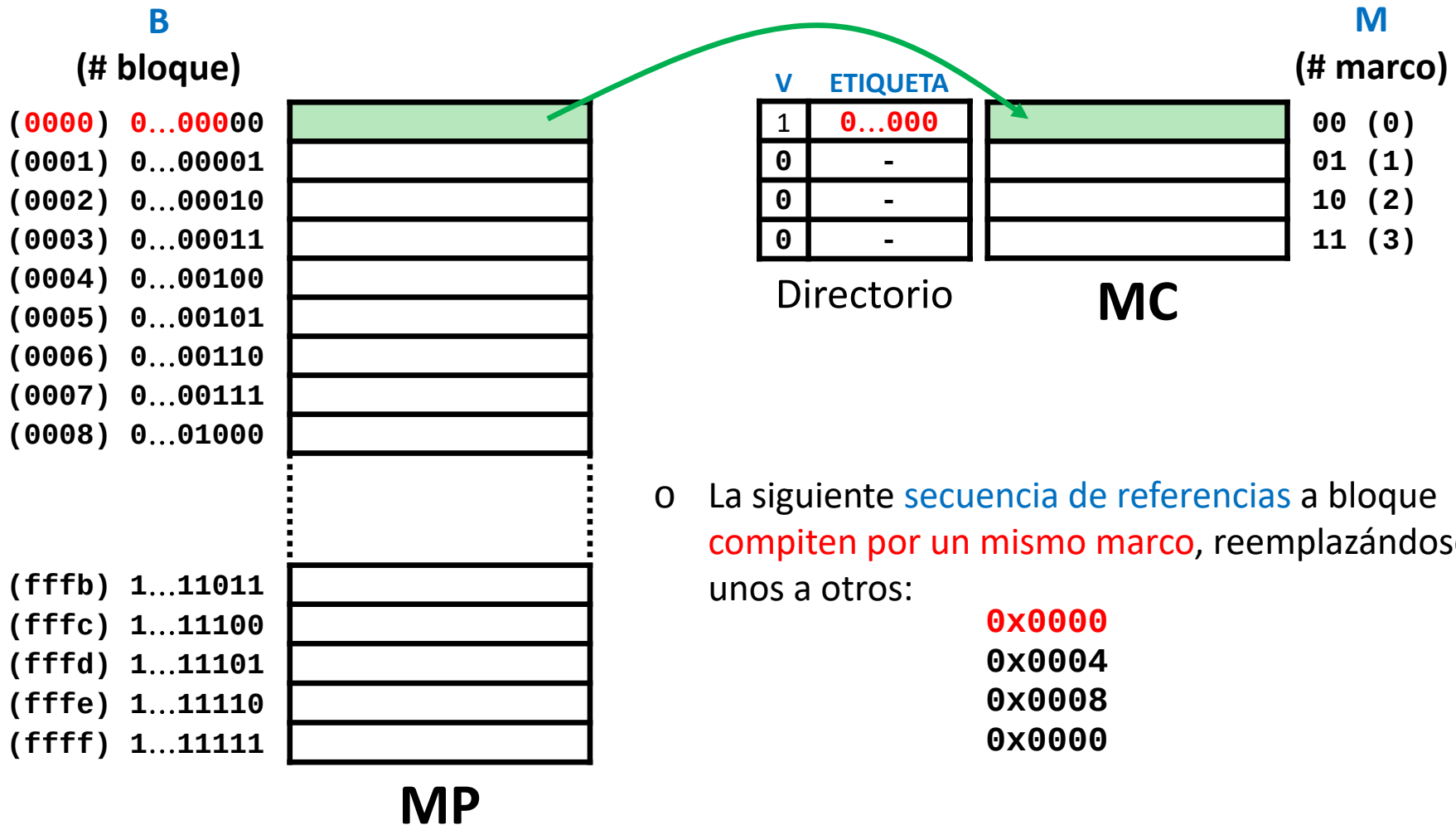
- La siguiente secuencia de referencias a bloque compiten por un mismo marco, reemplazándose unos a otros:

```
0x0000
0x0004
0x0008
0x0000
```

MC de emplazamiento directo

Ejemplo de fallos por conflicto (ii)

- Una MC con emplazamiento directo puede dar fallo aunque esté prácticamente vacía.



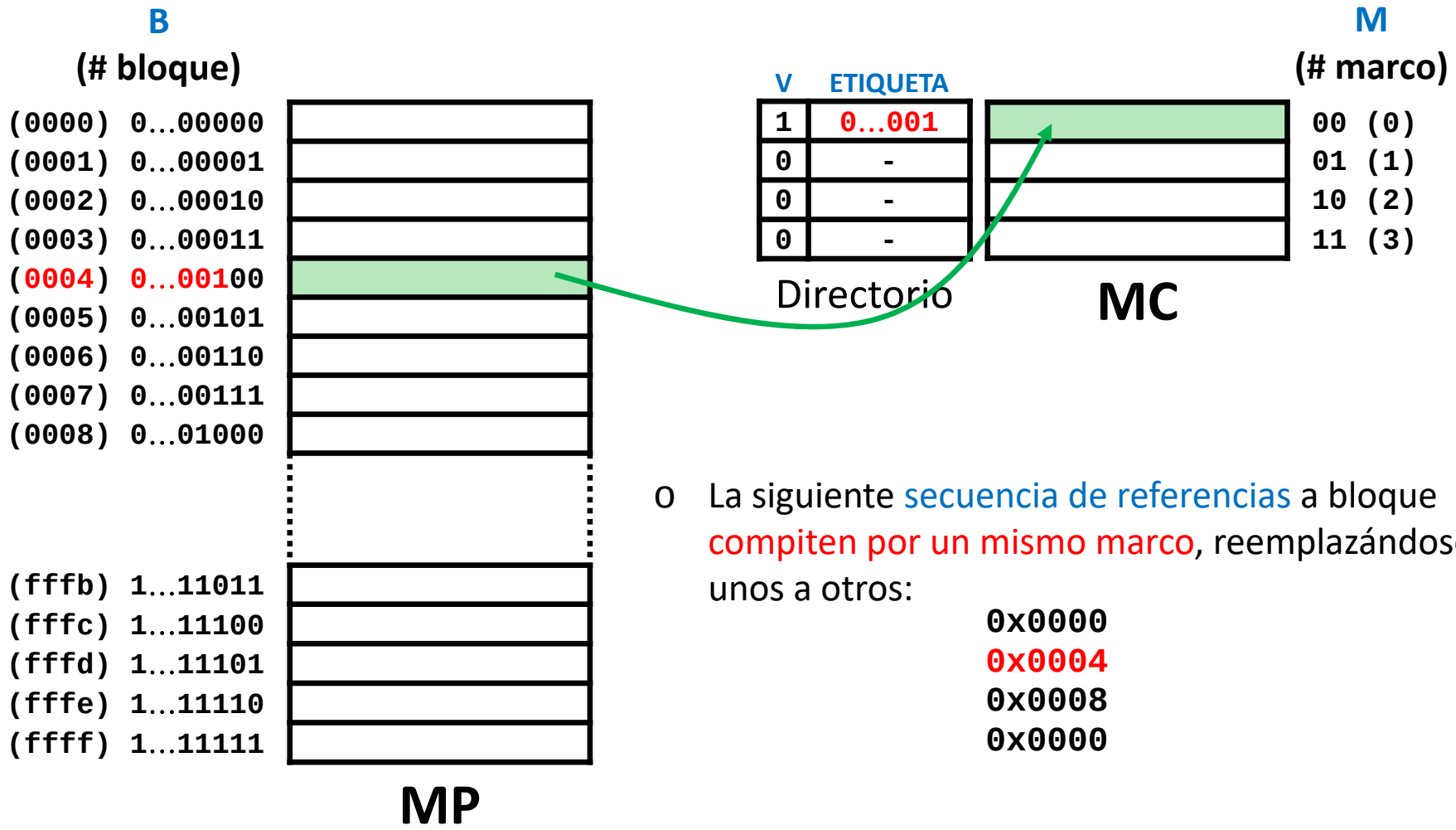
- La siguiente secuencia de referencias a bloque compiten por un mismo marco, reemplazándose unos a otros:

  0x0000
  0x0004
  0x0008
  0x0000

MC de emplazamiento directo

Ejemplo de fallos por conflicto (iii)

- Una MC con emplazamiento directo puede dar fallo aunque esté prácticamente vacía.



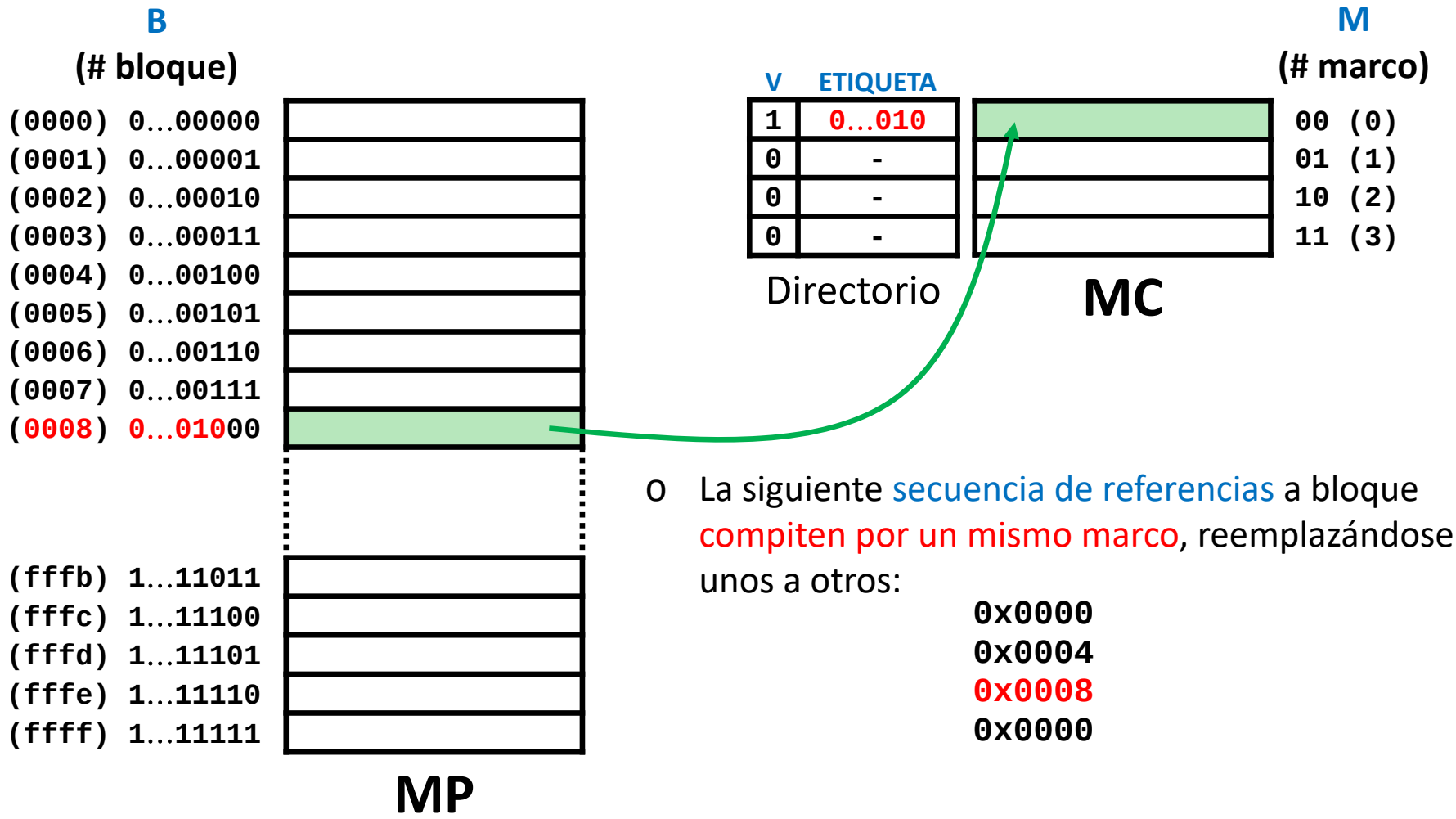
- La siguiente secuencia de referencias a bloque compiten por un mismo marco, reemplazándose unos a otros:

```
0x0000
0x0004
0x0008
0x0000
```

MC de emplazamiento directo

Ejemplo de fallos por conflicto (iv)

- Una MC con emplazamiento directo puede dar fallo aunque esté prácticamente vacía.



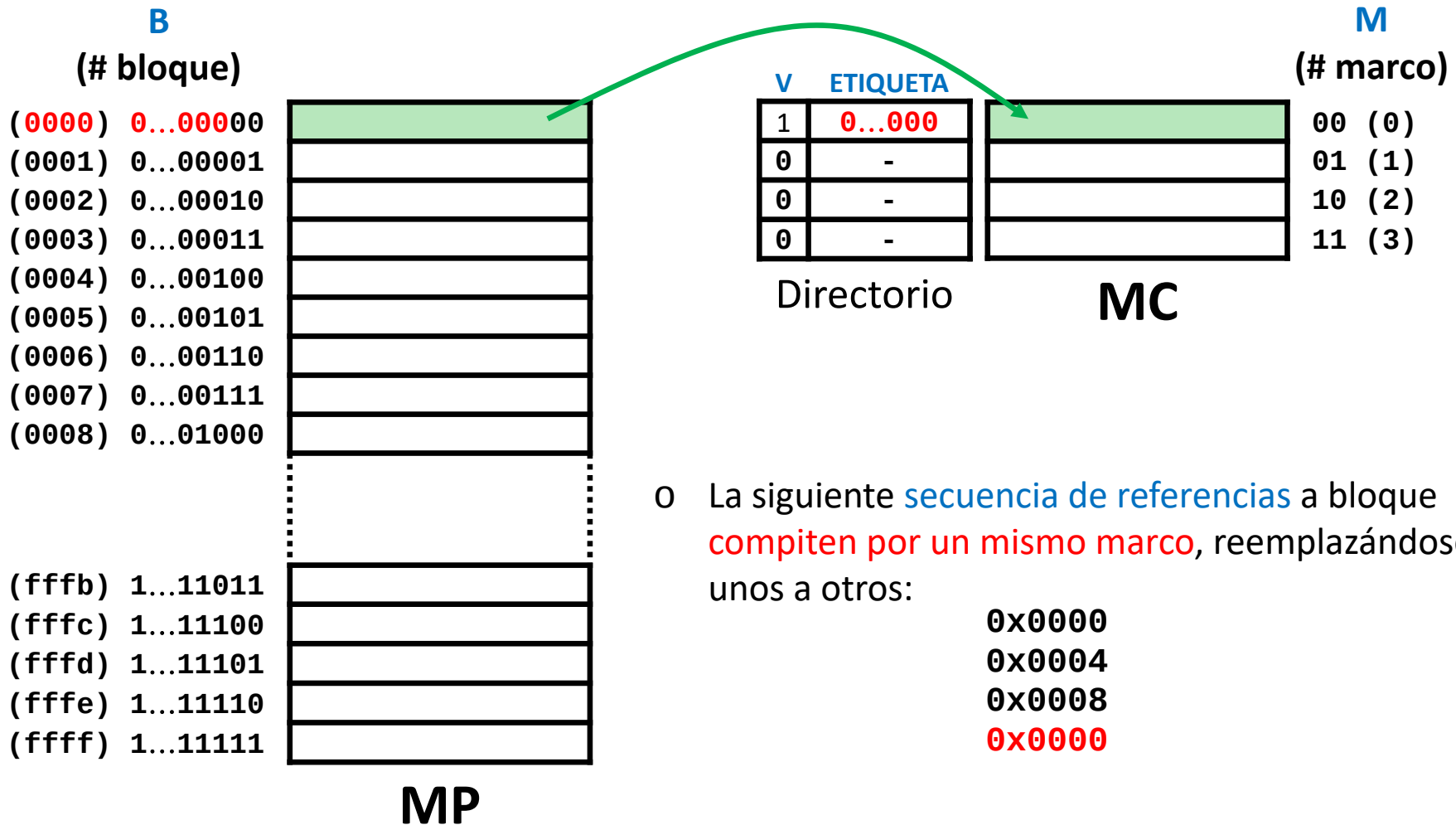
- La siguiente secuencia de referencias a bloque compiten por un mismo marco, reemplazándose unos a otros:

```
0x0000
0x0004
0x0008
0x0000
```

MC de emplazamiento directo

Fallos por conflicto

- Una MC con emplazamiento directo puede dar fallo aunque esté prácticamente vacía.



- La siguiente secuencia de referencias a bloque compiten por un mismo marco, reemplazándose unos a otros:

  0x0000
  0x0004
  0x0008
  0x0000

MC de emplazamiento directo

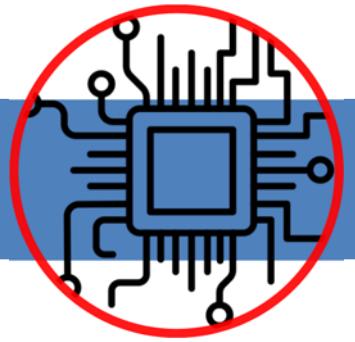
Conclusiones

- La MC con emplazamiento directo tiene numerosas ventajas:
  - Tiempos bajos de identificación, de acceso y de elección de marco.
    - Todo bloque tiene un marco único en donde puede cargarse.
    - El acceso al marco y al directorio es directo a partir de la dirección recibida.
    - Se requiere una única comparación para saber si el bloque está cargado.
  - Bajo coste hardware.
    - El directorio solo almacena parte de la dirección del bloque.
    - La identificación requiere un único comparador.

- Sin embargo, tiene una desventaja importante:
  - Alta probabilidad de fallos por conflicto.
    - Puede haber reemplazamientos aún estando la MC prácticamente vacía.
    - Suele tener una tasa de fallos superior a otras políticas.

- Celdas elementales de memoria RAM.
- Memoria SRAM asíncrona.
- Memoria DRAM asíncrona.
- Memoria DRAM síncrona.
- Módulos de memoria DRAM síncrona.
- Controlador de memoria.

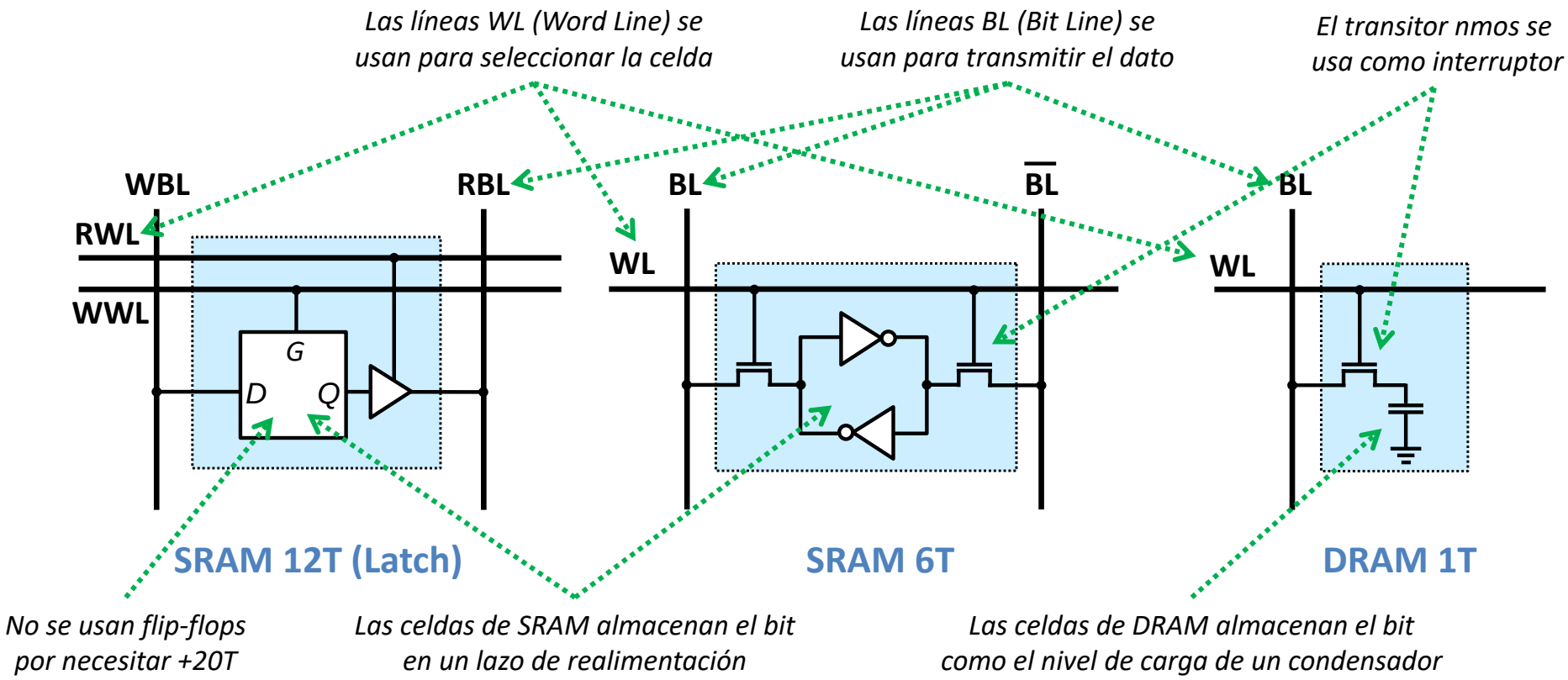
Apéndice tecnológico



Aspectos tecnológicos

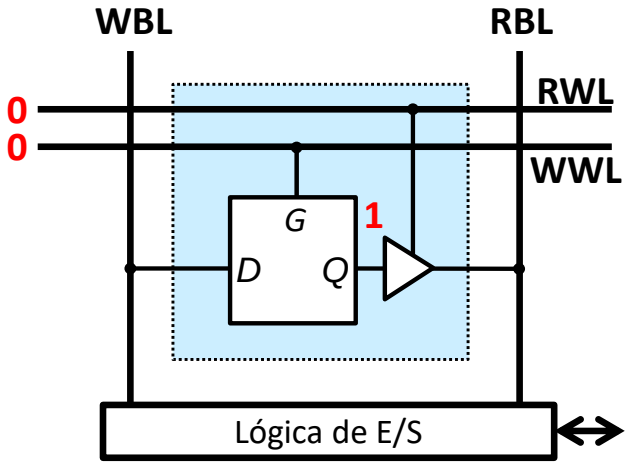
Celdas elementales en memorias RAM

- Las celdas de RAM se clasifican por:
  - El número de transistores usados en su implementación.
  - Por la tecnología usada para almacenar 1 bit de información.

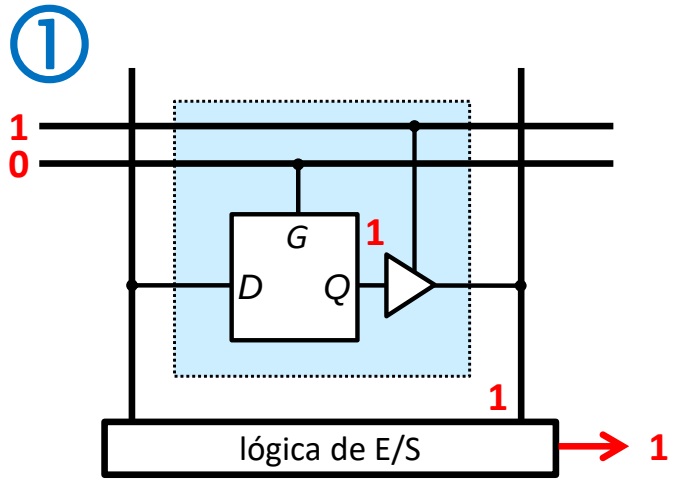


Aspectos tecnológicos

Celda SRAM 12T: lectura

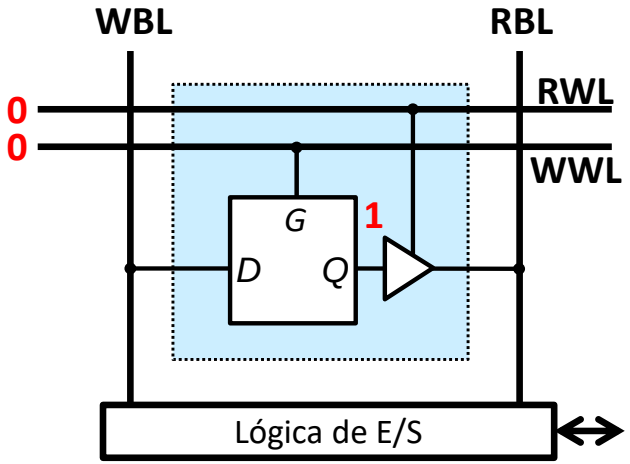


- Se activa RWL:
  - La celda propaga el valor almacenado por RBL.
  - La celda sigue conservando el valor almacenado.
- La operación finaliza desactivando RWL.

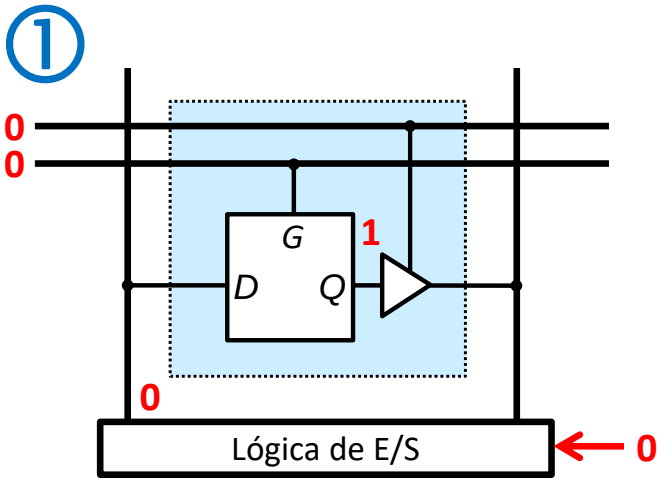


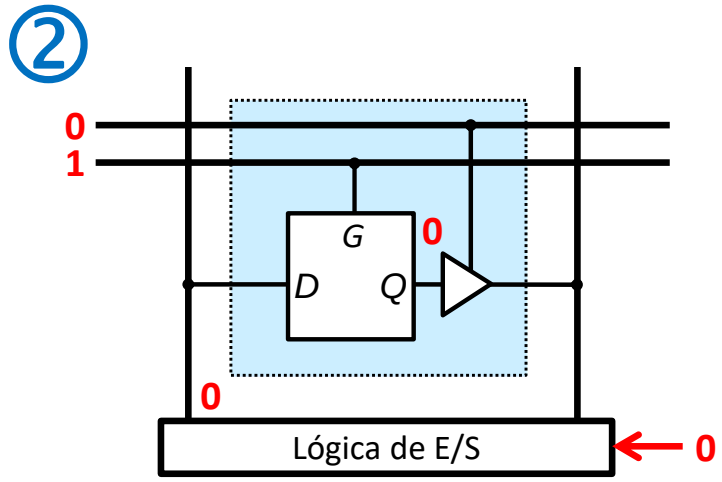
Aspectos tecnológicos

Celda SRAM 12T: escritura



- Se pone en WBL el valor a almacenar.
- Se activa WWL:
  - La celda carga el valor a almacenar.
  - Esta es una operación rápida.
- La operación finaliza desactivando WWL.

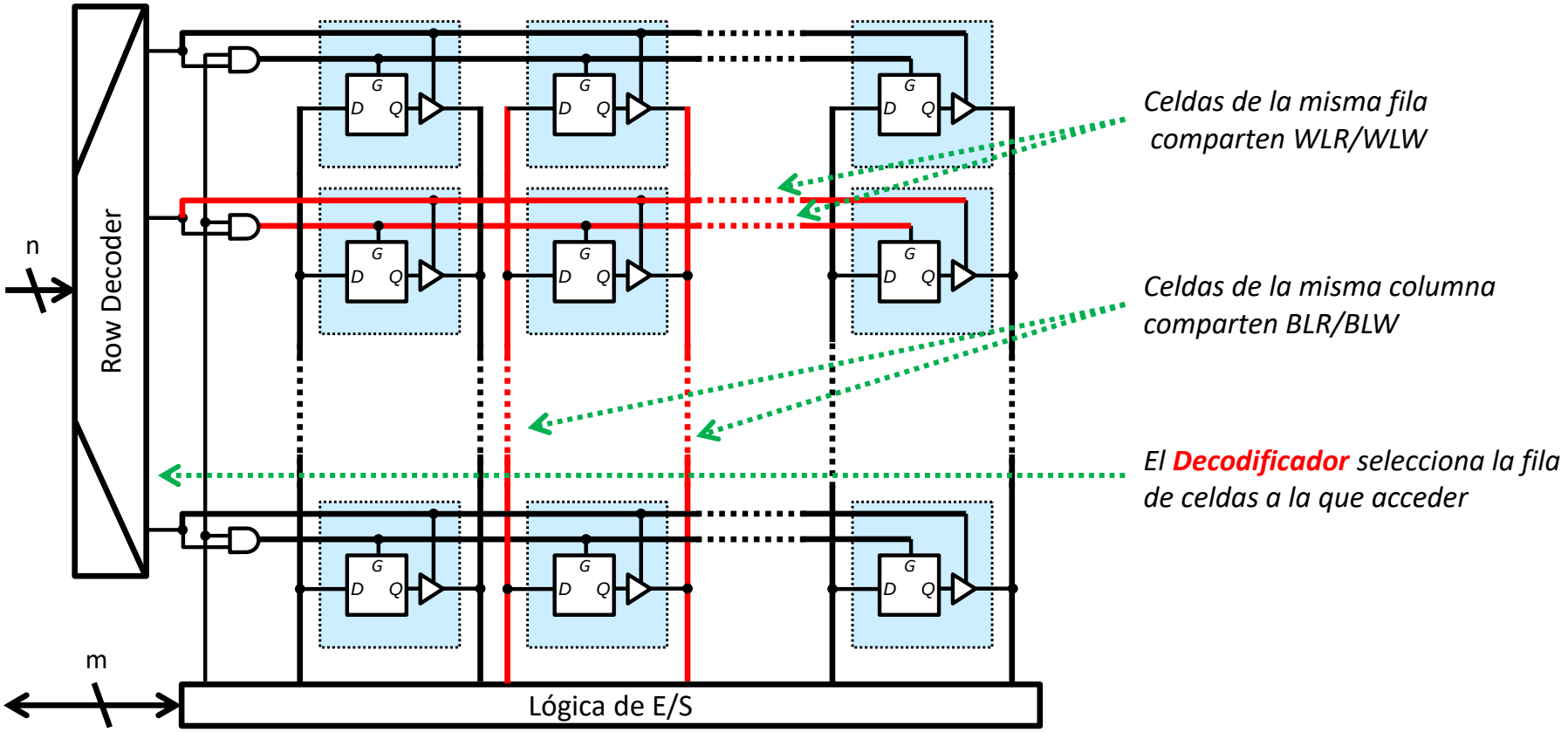




Aspectos tecnológicos

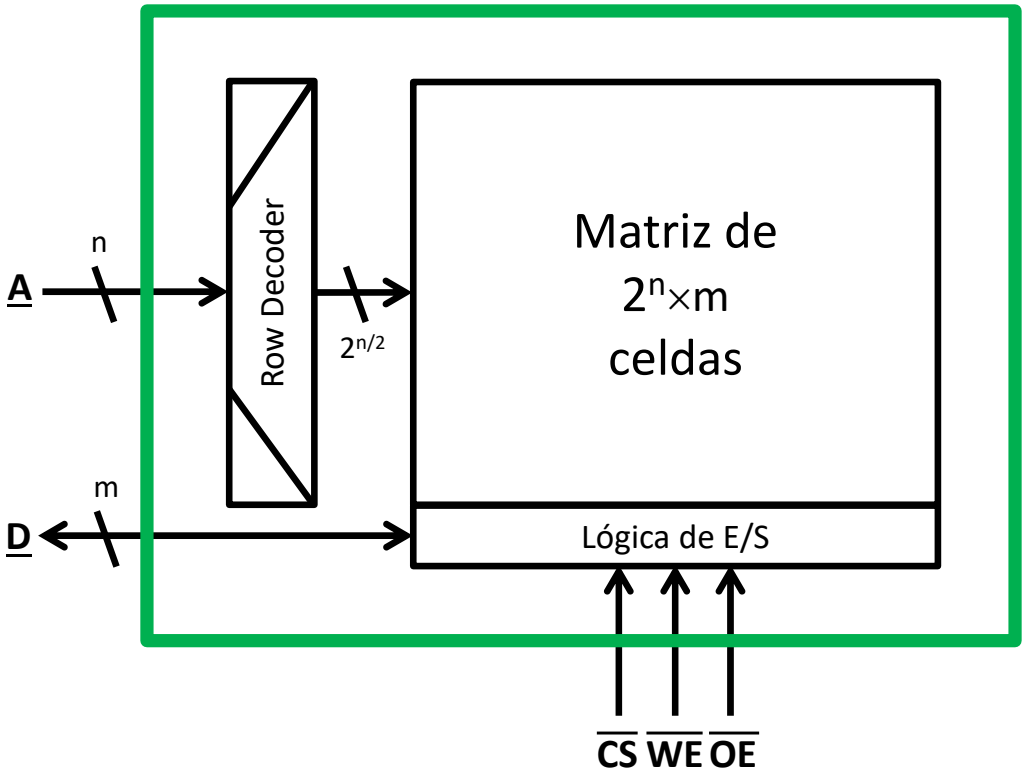
Memoria SRAM (12T) asíncrona: matriz de celdas

- Las celdas se distribuyen en una matriz $2^n \times m$ (con $2^n >> m$) con:
  - Tantas filas como palabras tenga la memoria.
  - Tantas columnas como anchura tenga la palabra.



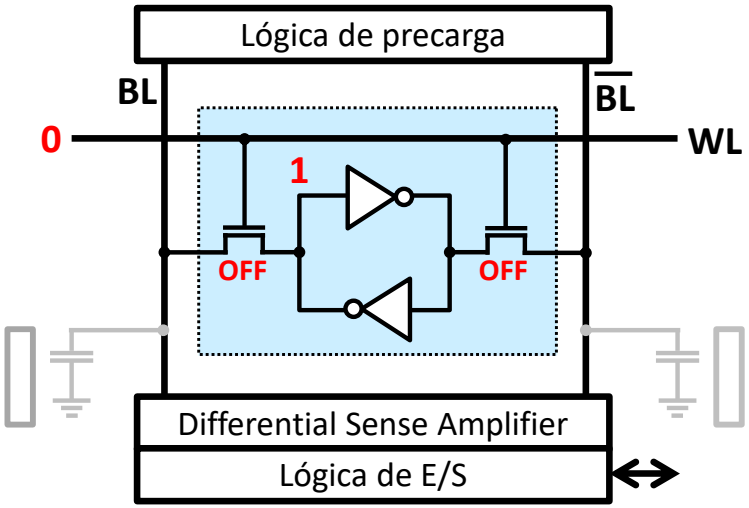
Aspectos tecnológicos

Memoria SRAM (12T) asíncrona: estructura



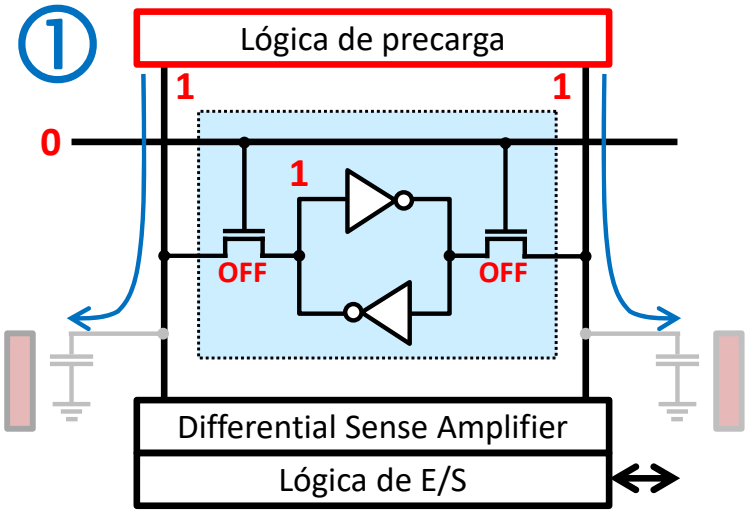
Aspectos tecnológicos

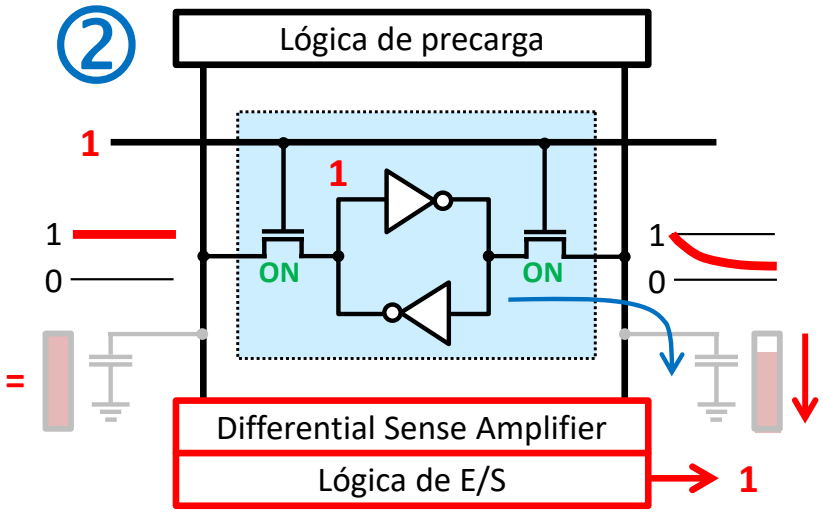
Celda SRAM 6T: lectura



1. Se precargan a Vdd las BL.
2. Se activan WL y *sense amplifier:*
   - La celda propaga los valores que almacena por las BLs.
   - La celda conserva el valor almacenado.

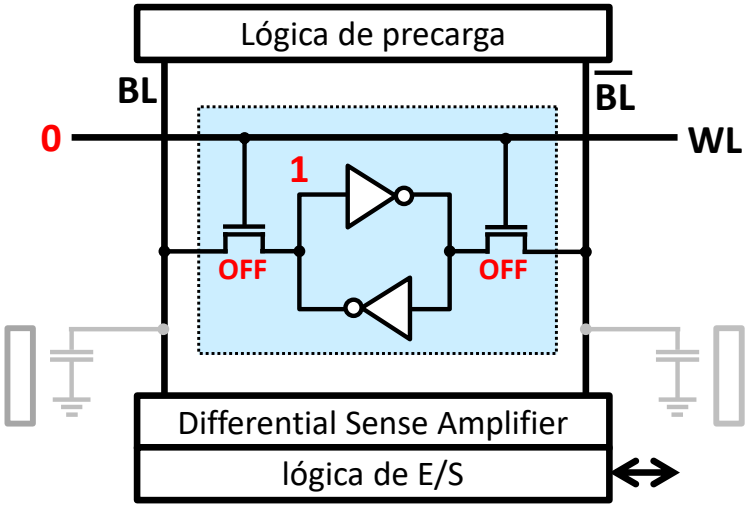
- Se amplifica la pequeña diferencia de voltaje entre BLs
- Se transfiere al exterior el valor leído.
- Se desactivan WL y *sense amplifier.*



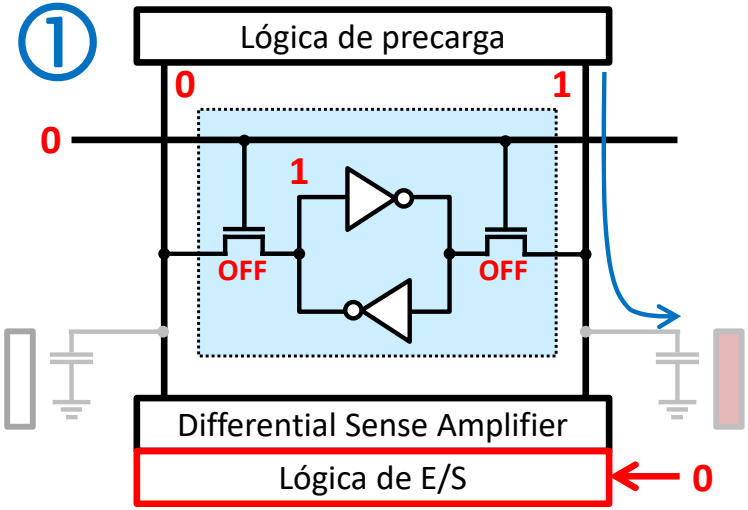


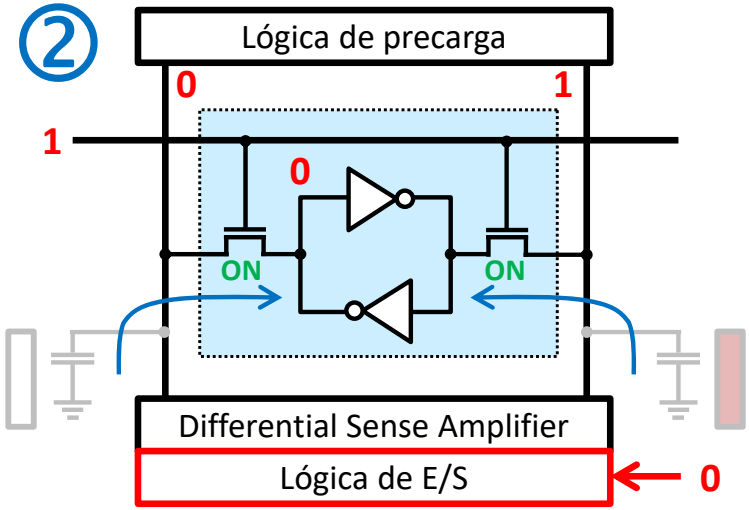
Aspectos tecnológicos

Celda SRAM 6T: escritura



1. El valor a almacenar se pone en BL e invertido en $\overline{BL}$.
2. Se activa WL:
   - La celda carga el valor a almacenar.
- Se desactiva WL.

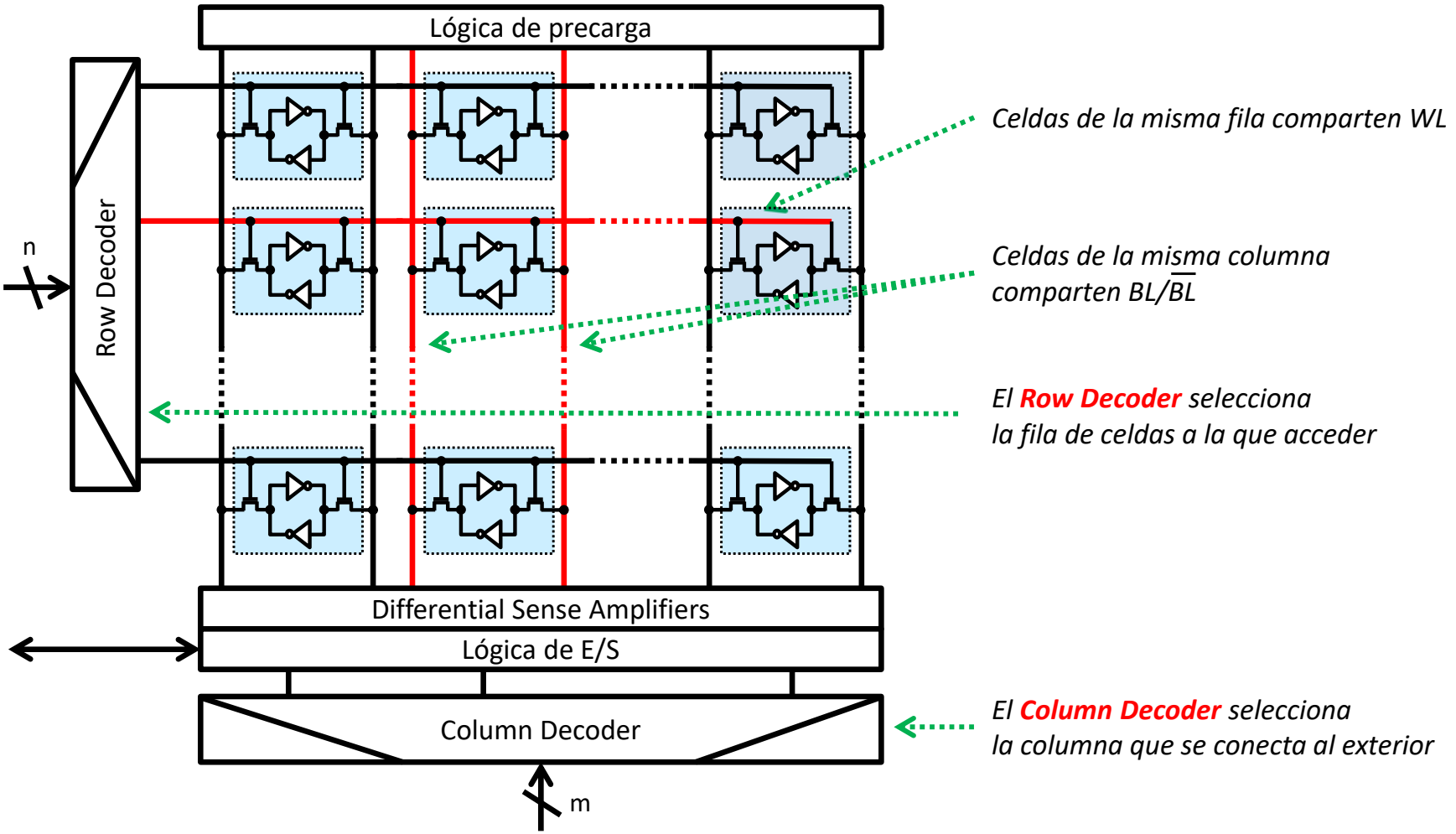




Aspectos tecnológicos

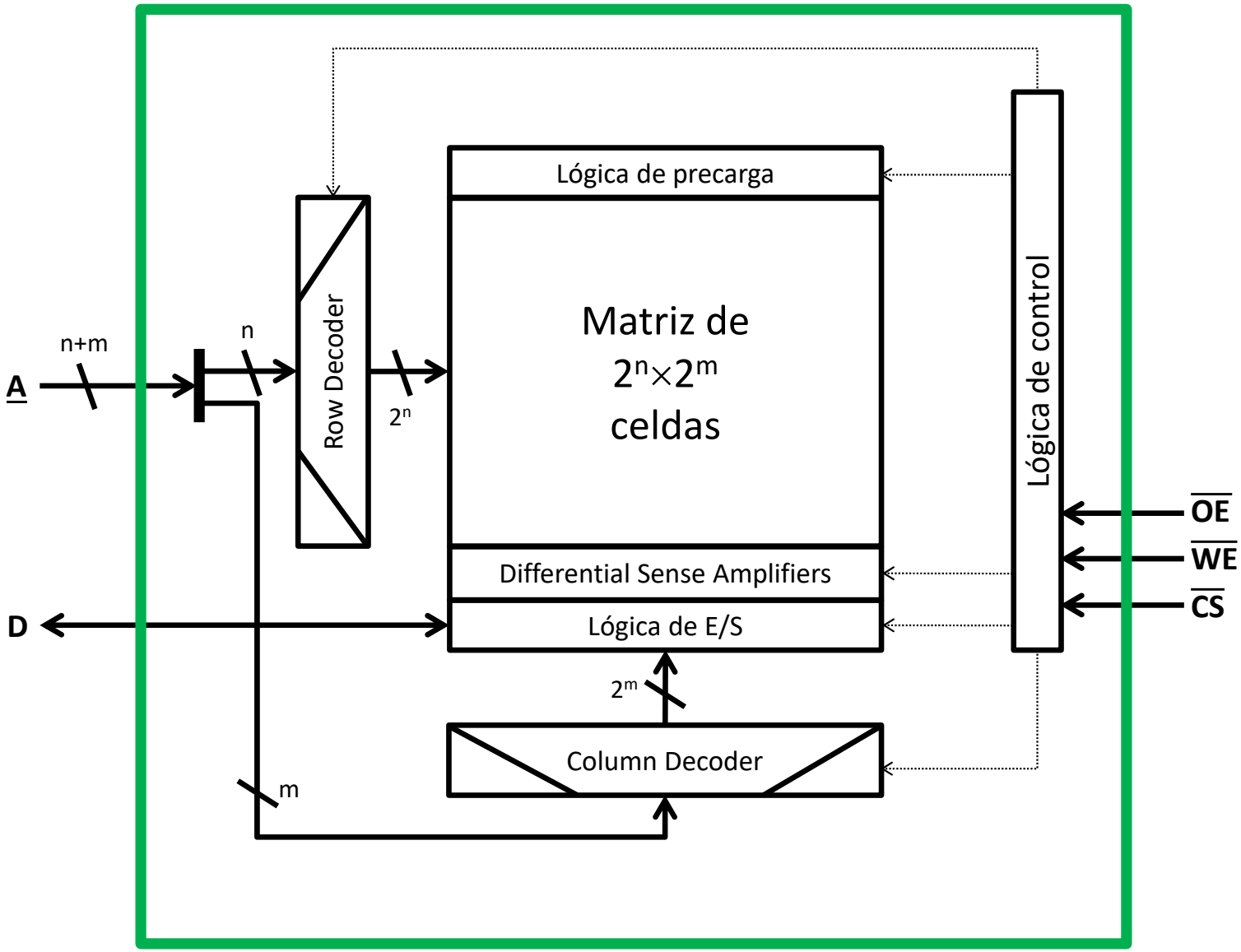
Memoria SRAM (6T) asíncrona: matriz de celdas

- Las celdas se distribuyen en una matriz $2^n \times 2^m$ (con $n \approx m$)
  - El decodificador se divide en 2 para reducir sustancialmente su coste.



Aspectos tecnológicos

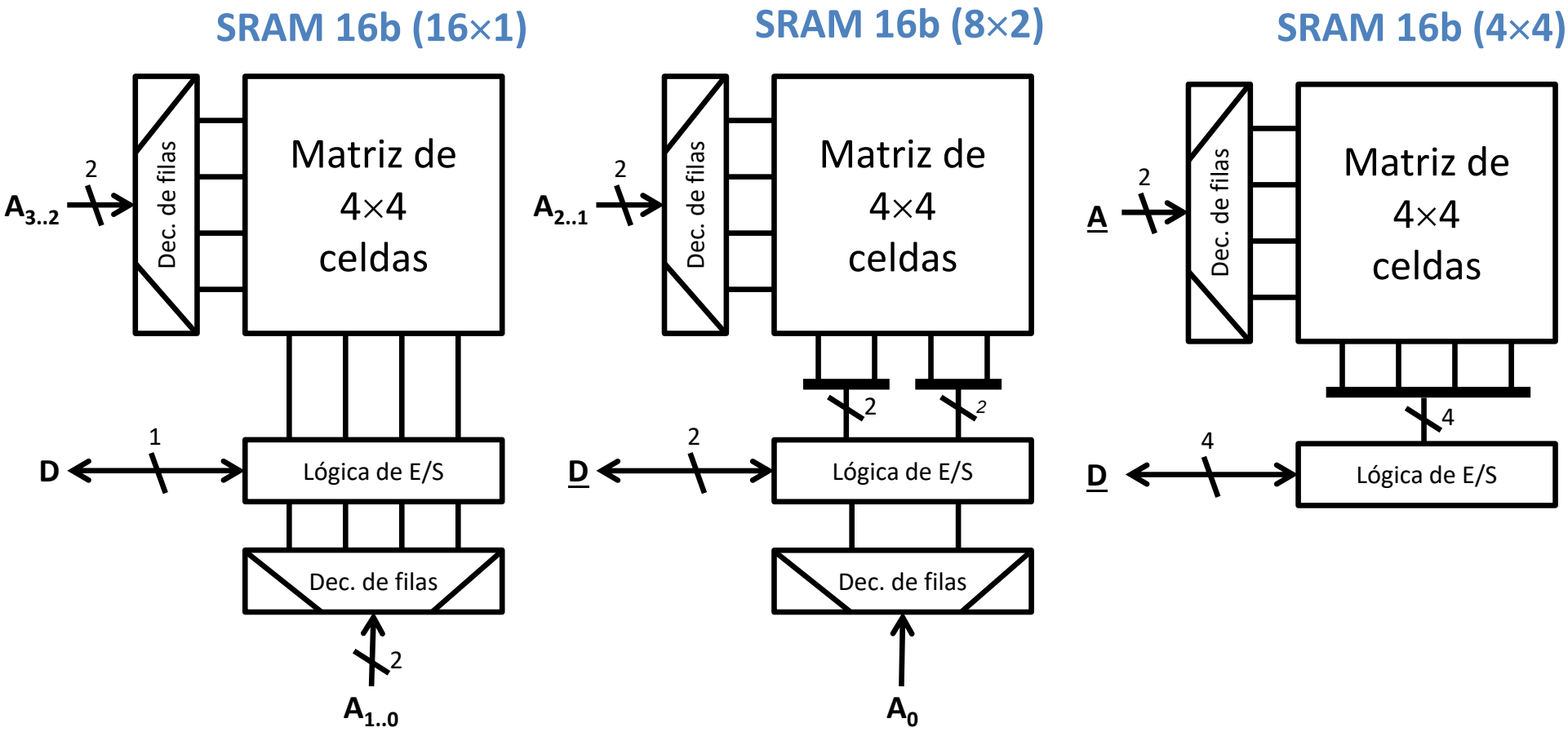
Memoria SRAM (6T) asíncrona: organización (i)



Aspectos tecnológicos

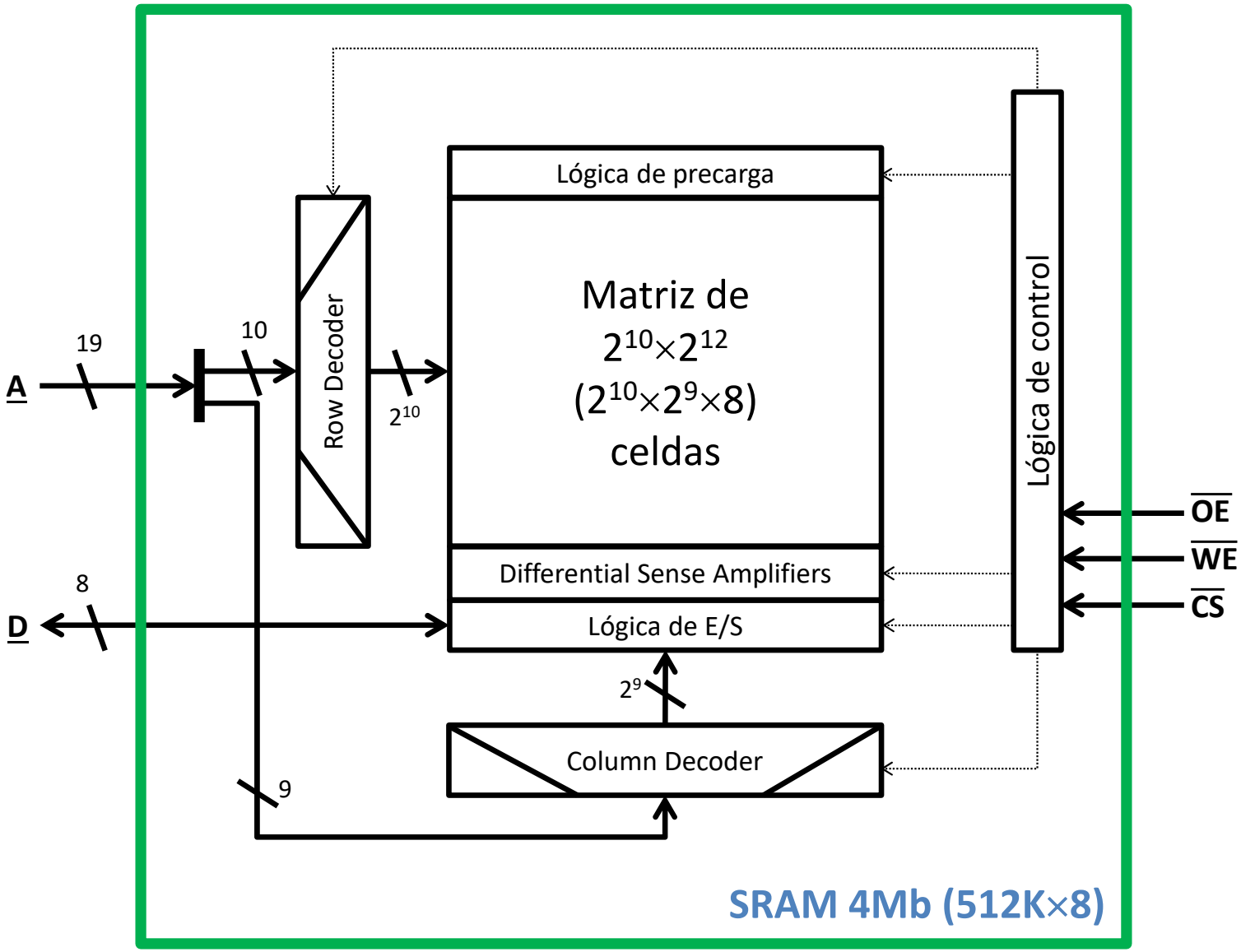
Memoria SRAM asíncrona (6T): organización (ii)

- Una misma organización física de celdas puede presentarse al exterior con diferentes organizaciones lógicas:
  - Por ejemplo, toda SRAM de 16b se diseñaría con una matriz de 4x4 celdas.



Aspectos tecnológicos

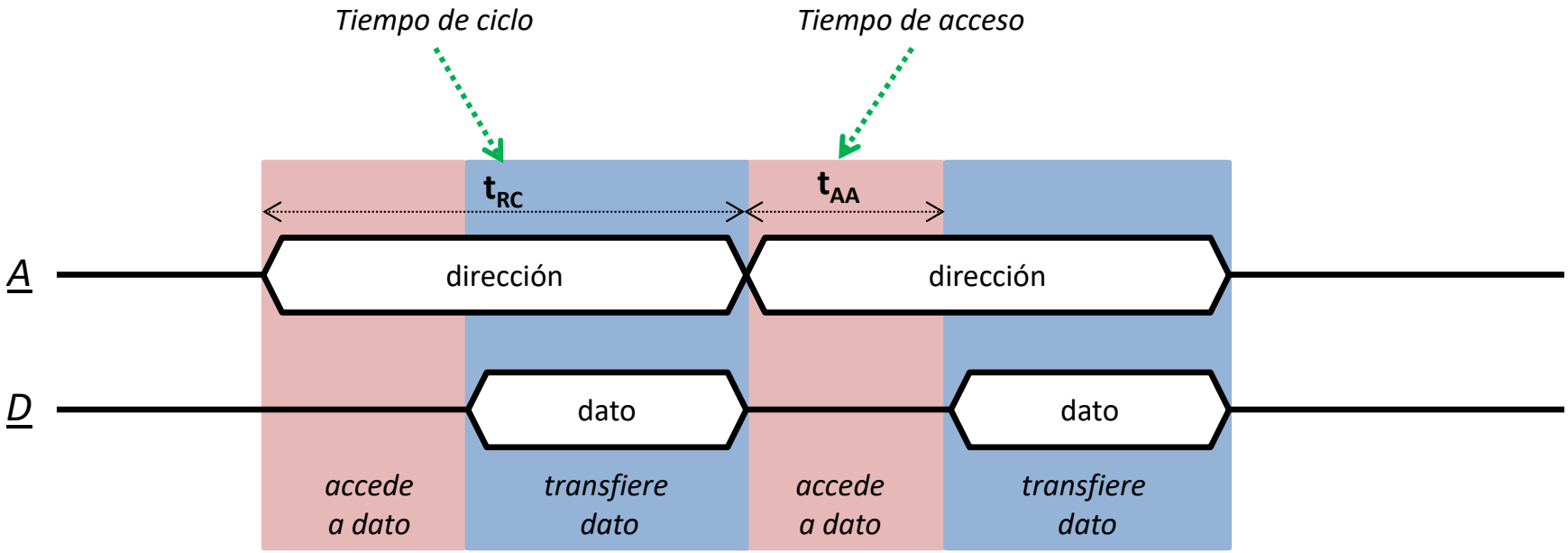
Memoria SRAM (6T) asíncrona: organización (iii)



Aspectos tecnológicos

Ciclo de acceso SRAM asíncrona (1964)

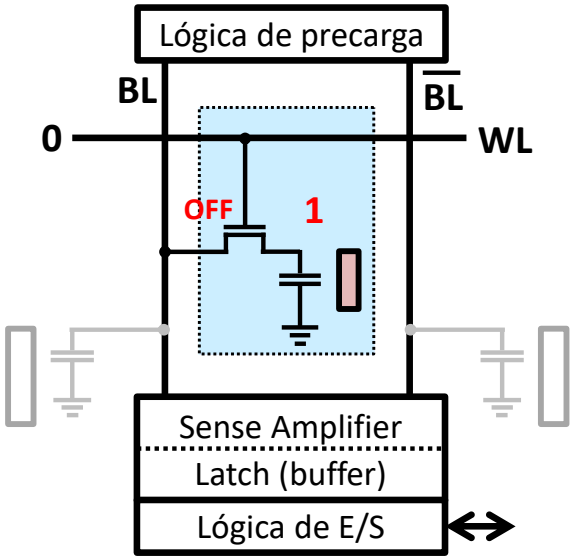
- SRAM (asíncrona): para acceder un dato basta con indicar su dirección.
  - Con las señales $\overline{OE}$ y $\overline{WE}$ (o R/$\overline{W}$) se indica si el acceso es de lectura o de escritura.



Ciclo de acceso simplificado

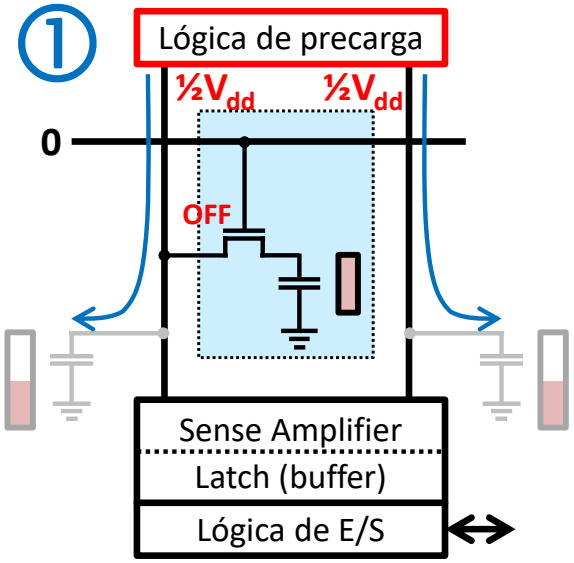
Aspectos tecnológicos

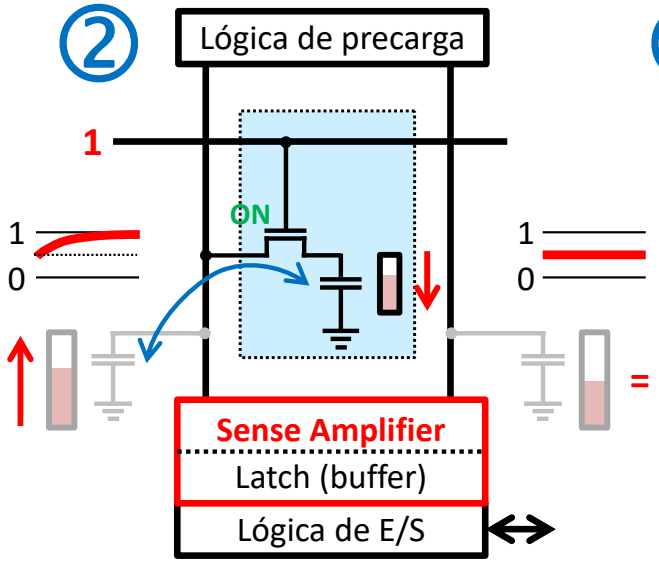
Celda DRAM 1T: lectura

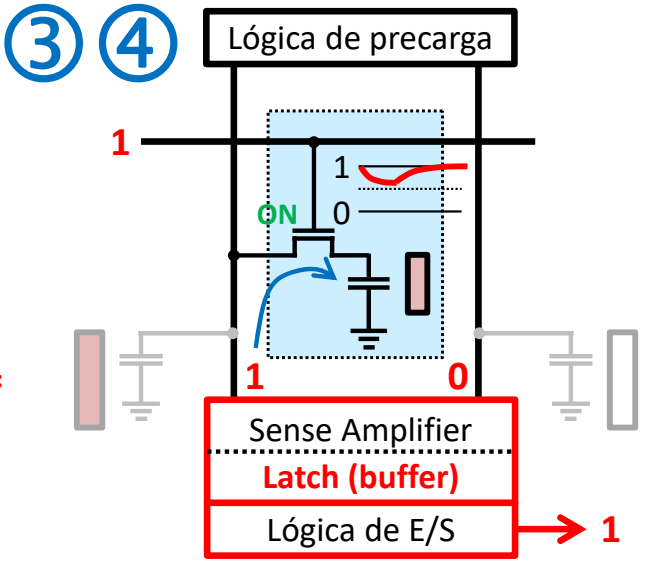


1. Precharge: Se precargan a ½Vdd ambas BL
2. Activate (Row Access): Se activan WL y *sense amplifier:*
   - La celda comparte carga con BL destruyendo el valor almacenado.
3. Se amplifica la pequeña diferencia de voltaje entre ambas BL.
   - Restaurando en la celda el valor leído.
4. Read (Column Access): Se transfiere al exterior el valor leído
   - Este valor se conserva en el buffer del amplificador.

- Se desactivan WL y *sense amplifier.*

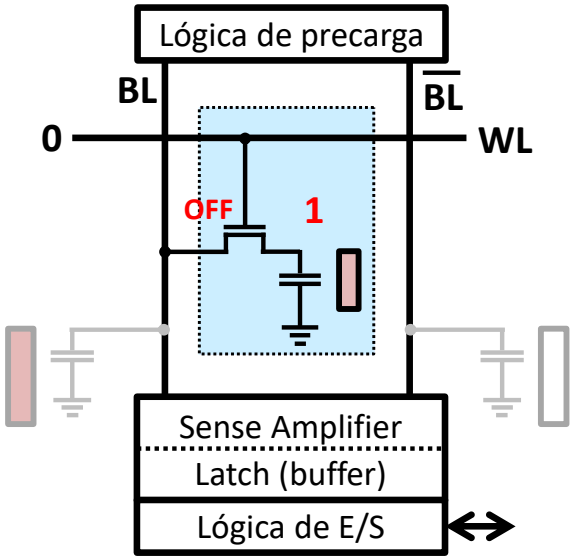






Aspectos tecnológicos

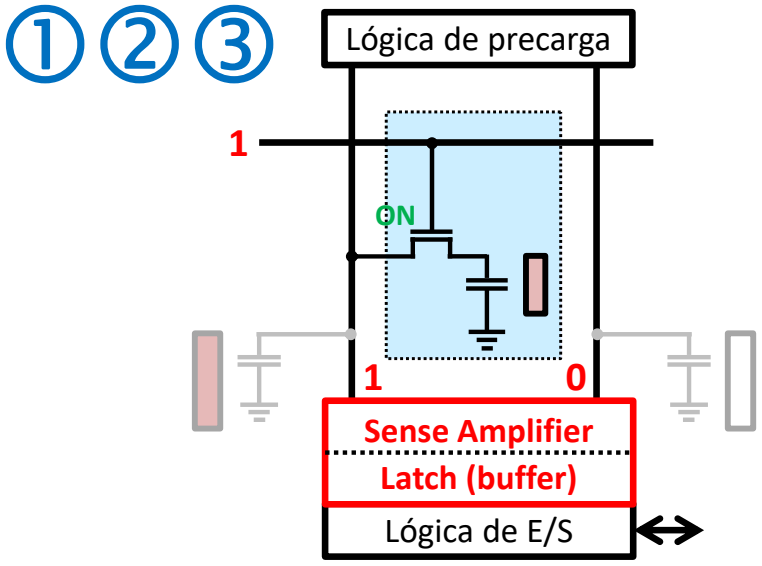
Celda DRAM 1T: escritura

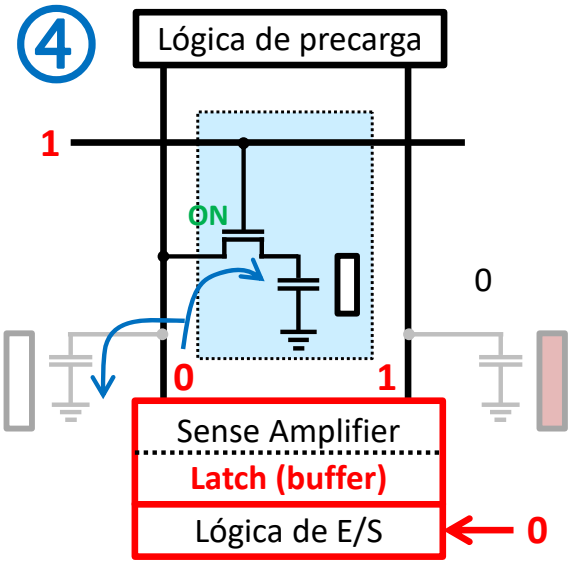


- Comienza como una lectura (Precharge+Activate).
  - Pero el valor leído no se transfiere al exterior.

4. Write (Column Access): El valor a almacenar se sobrescribe en el buffer del amplificador.

- El valor almacenado en el buffer se propaga por la BL:
  - La celda se carga/descarga en función del valor a almacenar.
- Se desactivan WL y *sense amplifier.*

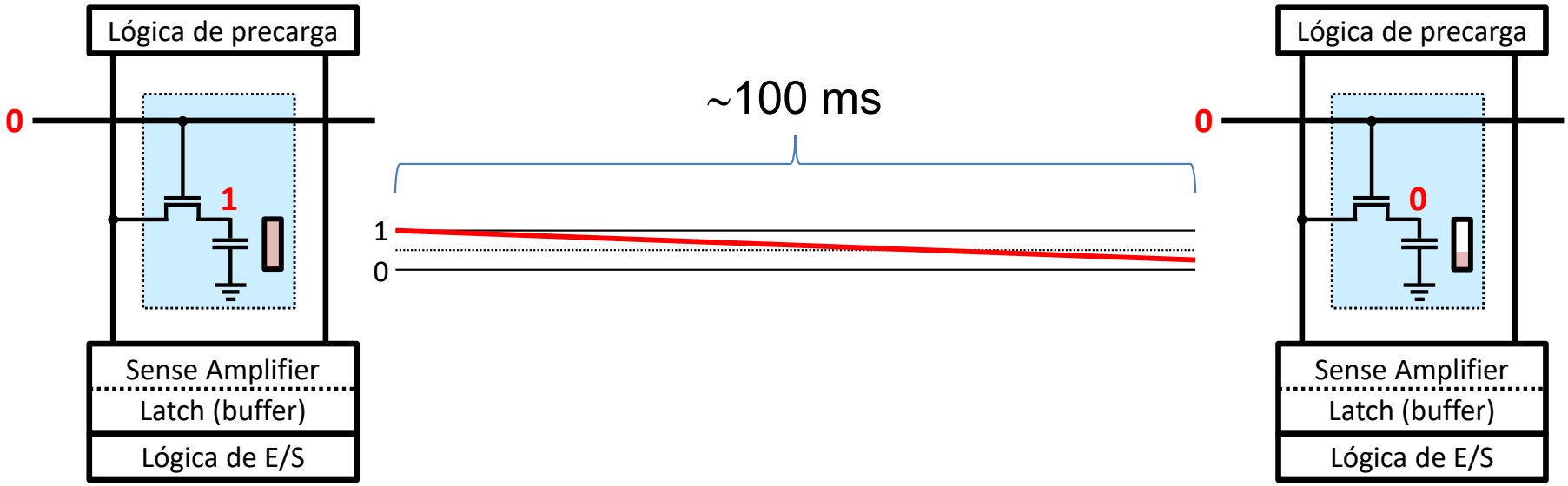




Aspectos tecnológicos

Celda DRAM 1T: refresco

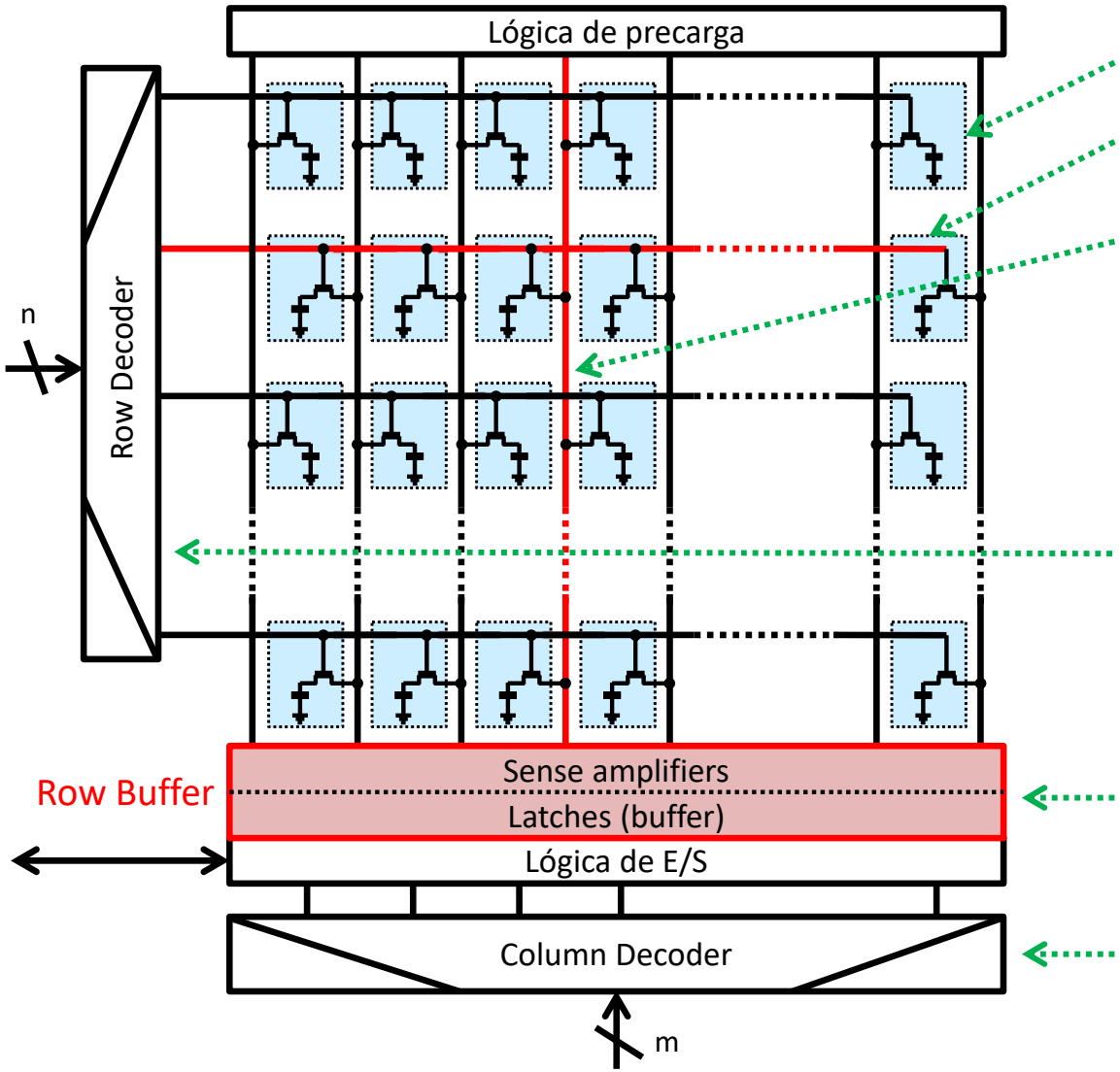
- La celda si no se lee/escribe, pierde gradualmente su carga.
  - Los 1 se convierten en 0 transcurrido cierto tiempo de inactividad.



- Para evitarlo, toda celda deber ser refrescada periódicamente (64 ms)
  - Dado que toda lectura reestablece el valor de la celda, el refresco se hace mediante lecturas sin transferencia al exterior (Precharge+Activate)

Aspectos tecnológicos

Memoria DRAM: matriz de celdas



Las celdas se distribuyen en una matriz 2D

Celdas de la misma fila comparten WL

Celdas alternas de la misma columna comparten BL

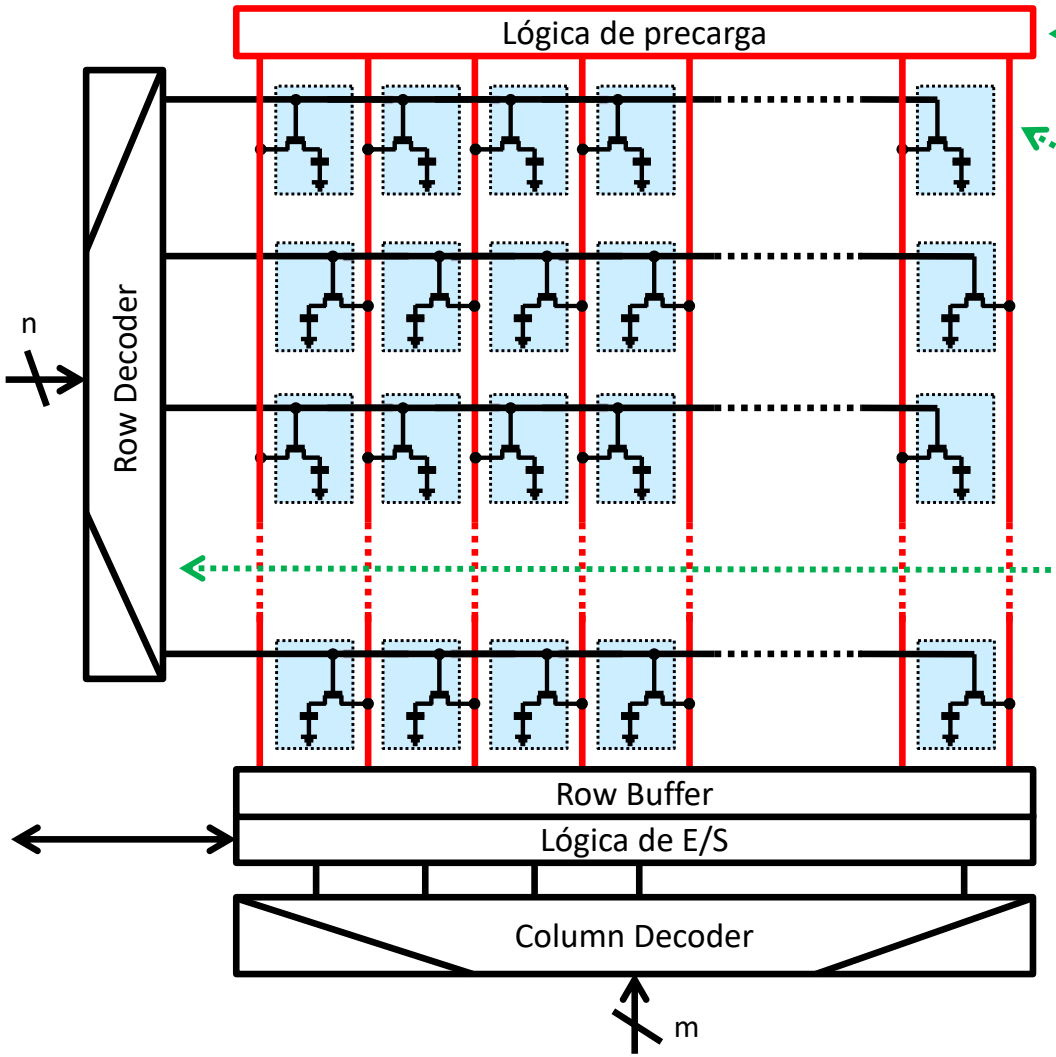
*El **Row Decoder** selecciona la fila de celdas a la que acceder*

*El **Row Buffer** carga todas las celdas de la fila seleccionada*

*El **Column Decoder** selecciona la celda del Row Buffer que se conecta al exterior*

Aspectos tecnológicos

Memoria DRAM: Precharge



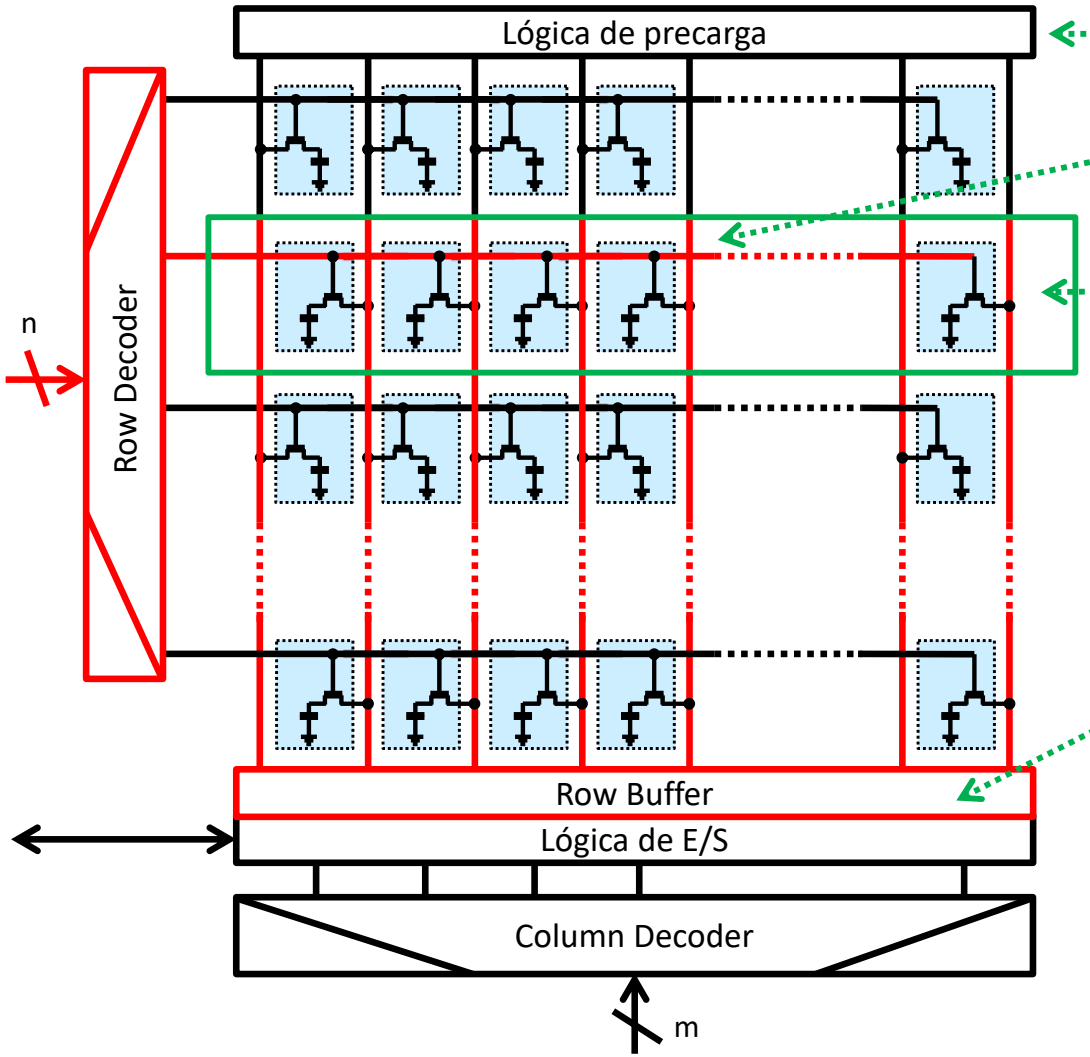
Se habilita la lógica de precarga

Se precargan todas las BL preparando la matriz para acceder a una de sus filas

Se deshabilita el Row Decoder para que todas las filas estén "cerradas"

Aspectos tecnológicos

Memoria DRAM: Activate (Row Access)



Se deshabilita la lógica de precarga

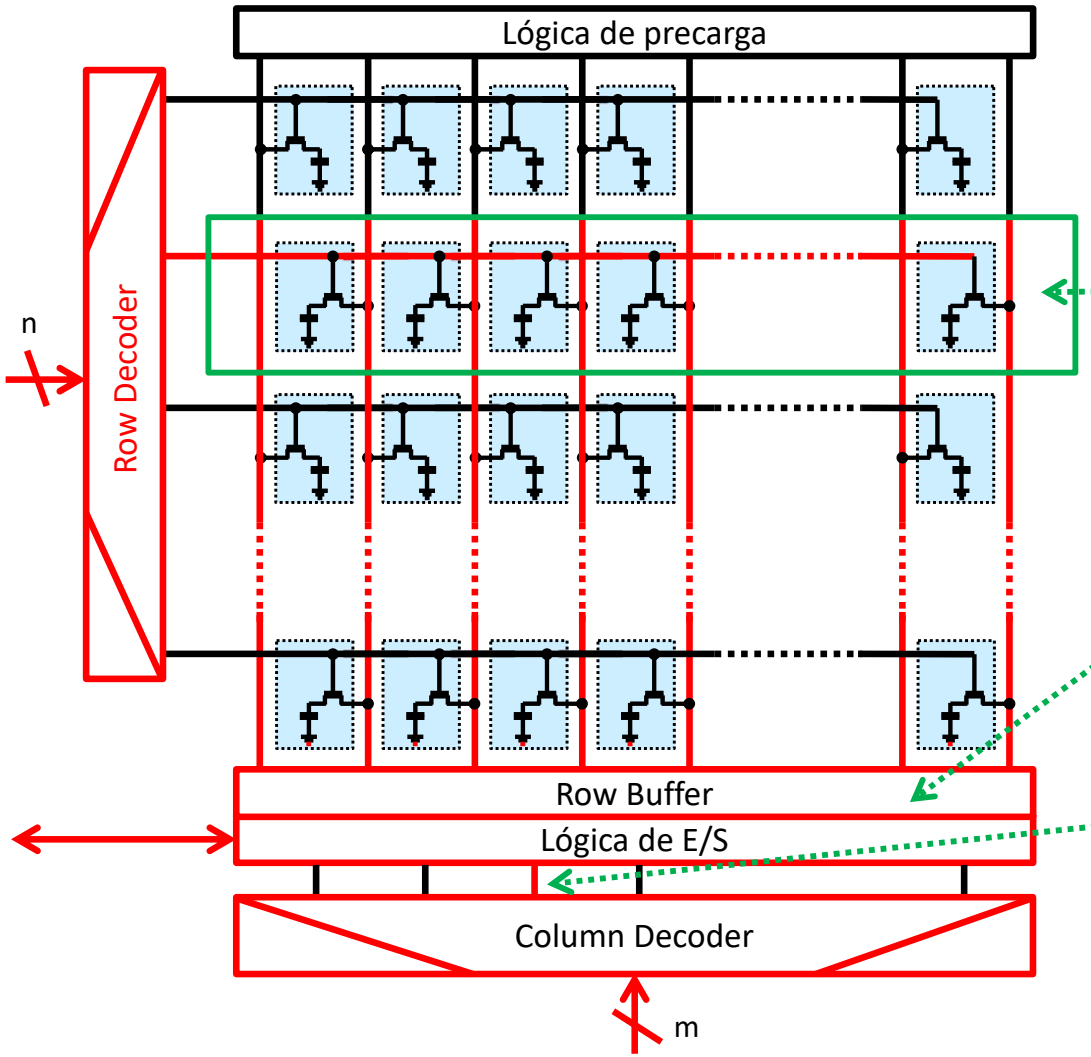
Se habilita el Row Decoder para que decodifique la dirección de fila y active la WL correspondiente

La fila seleccionada se "abre" conectándose al Row Buffer

Se activa el Row Buffer para que amplifique, restaure y almacene el valor todas las celda de la fila "abierta"

Aspectos tecnológicos

Memoria DRAM: Read / Write (Column Access)



La fila permanece "abierta" hasta que una nueva precarga la "cierre".

Mientras la fila esté "abierta" puede accederse a cualquiera de sus bits

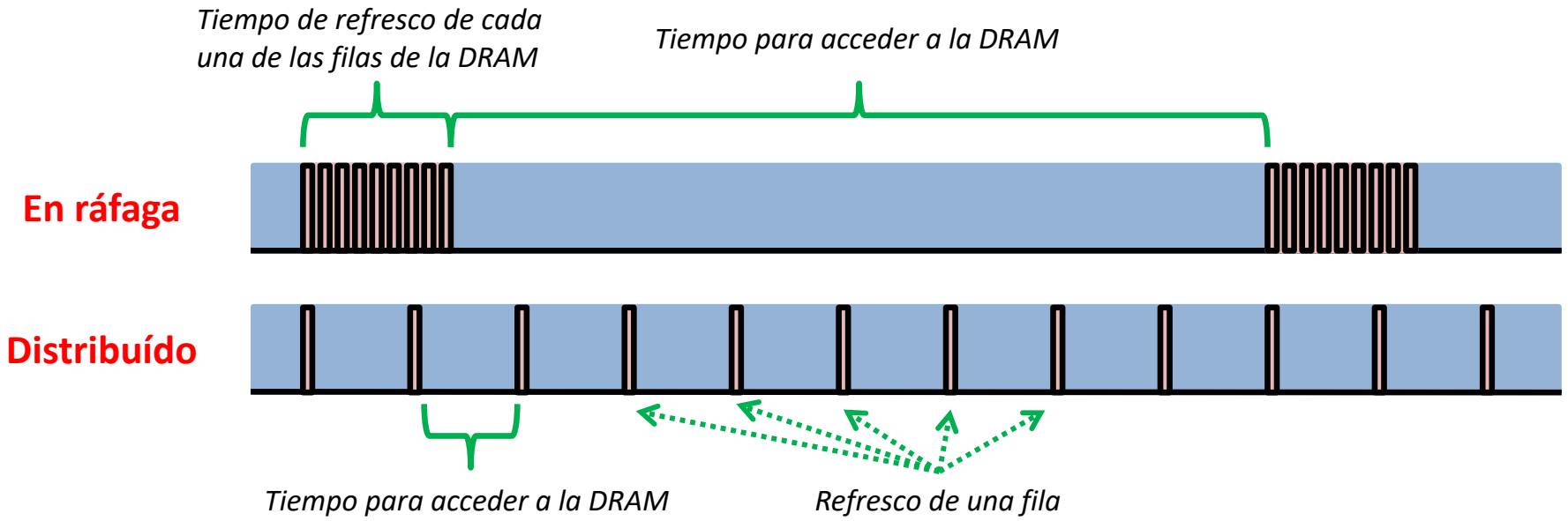
Se activa el Column Decoder para seleccionar un bit (o varios) del Row Buffer y:

- *Enviarlo al exterior (Lectura)*
- *Modificarlo en el buffer y en la fila (Escritura)*

Aspectos tecnológicos

Memoria DRAM: Refresh

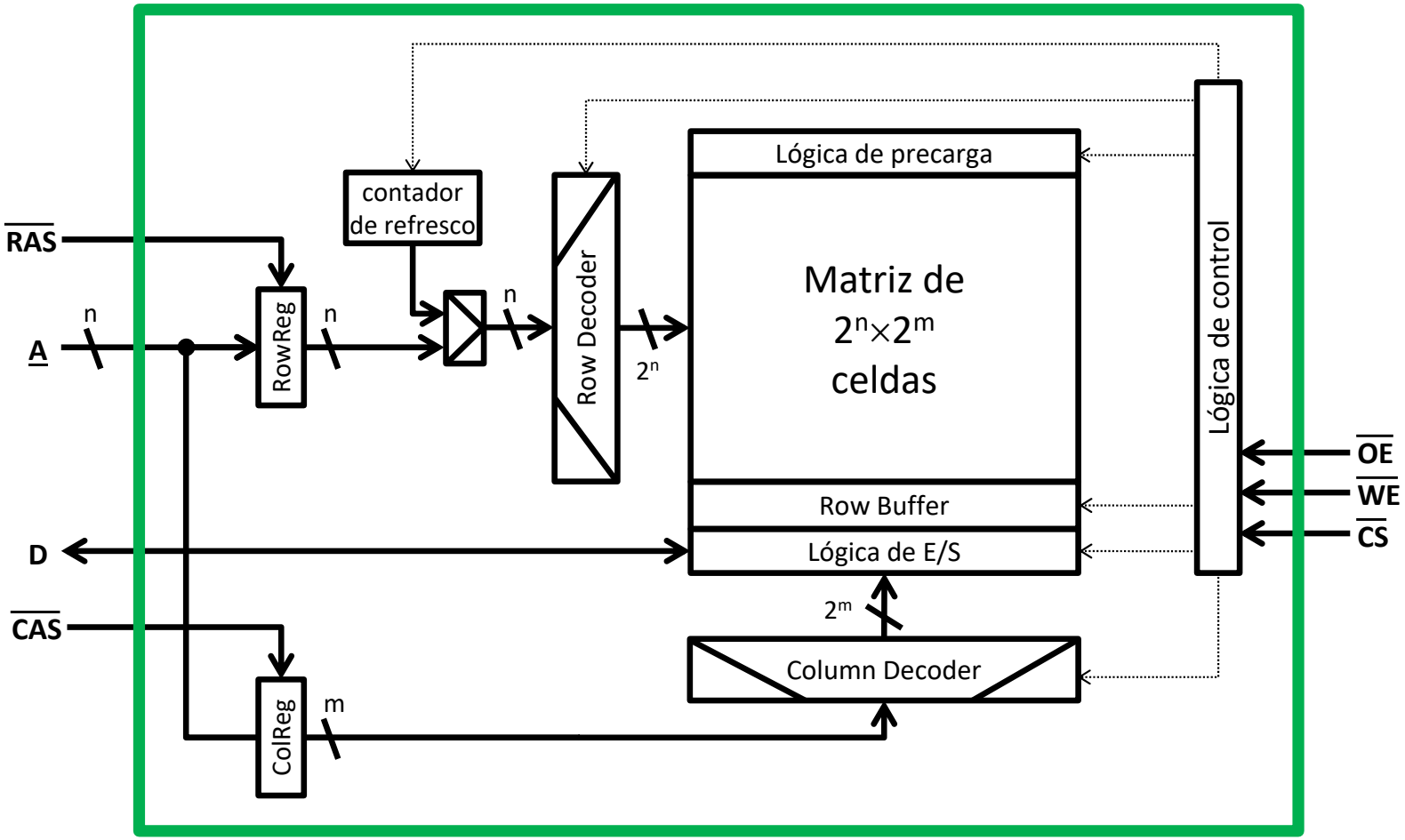
- El refresco se realiza por filas mediante ciclos de Precarga+Activate que lanza la propia memoria (*Seft-Refresh*) o un controlador externo.
- El refresco de la memoria completa puede hacerse:
  - En ráfaga: Todas las filas de la DRAM se refrescan consecutivamente.
    - La memoria está inaccesible durante un largo periodos de tiempo.
  - Distribuido: Se espacia el refresco de cada fila (o grupo de ellas).
    - Los accesos deben intercalarse entre los ciclos de refresco.



Aspectos tecnológicos

Memoria DRAM asíncrona: organización

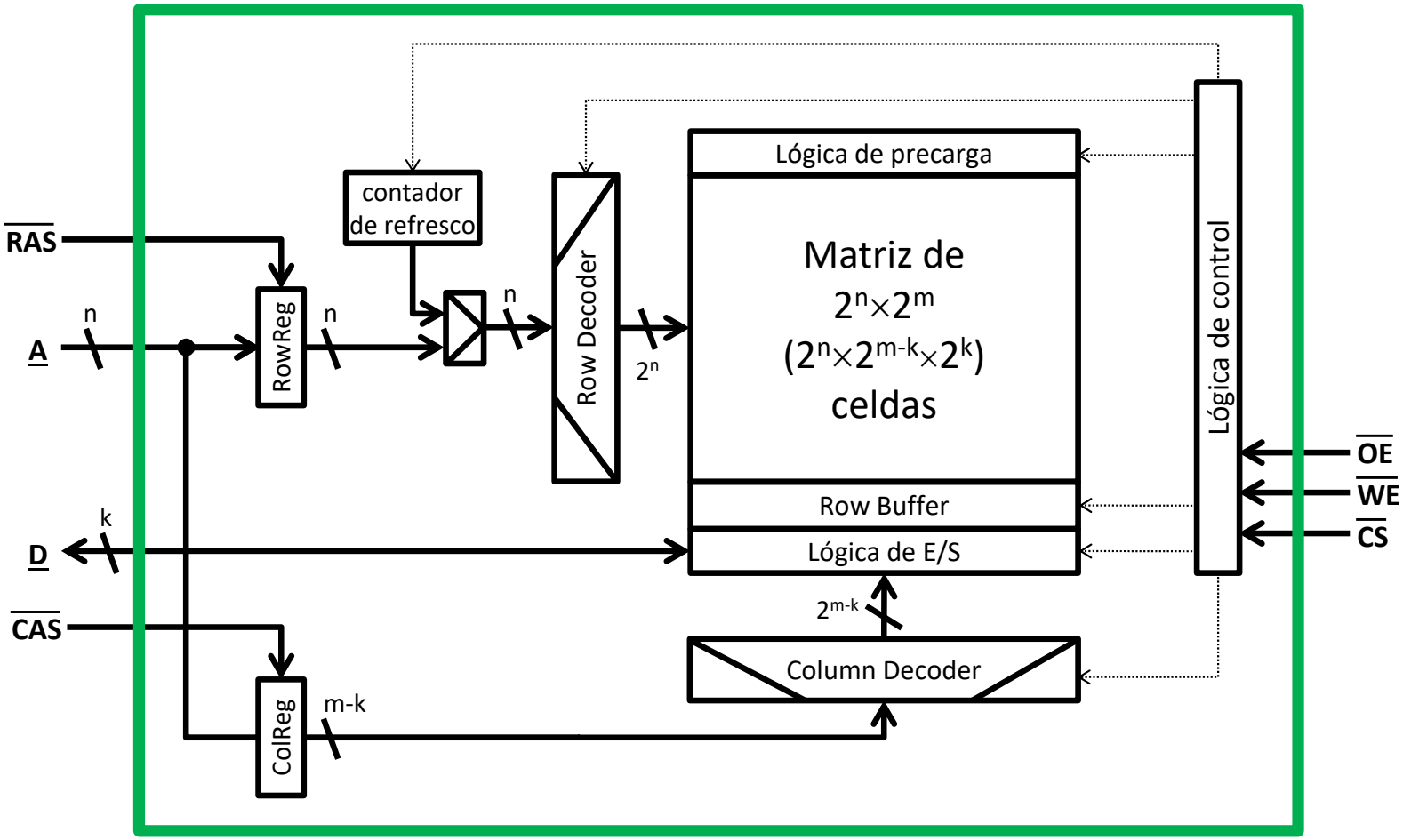
- Las celdas se distribuyen en una matriz $2^n \times 2^m$ (con $n \approx m$)
  - n (fila) y m (columna) se envían en instantes distintos por los mismos pines.
  - n y m se obtienen de fragmentar en 2 la dirección enviada por la CPU.



Aspectos tecnológicos

Memoria DRAM asíncrona: organización

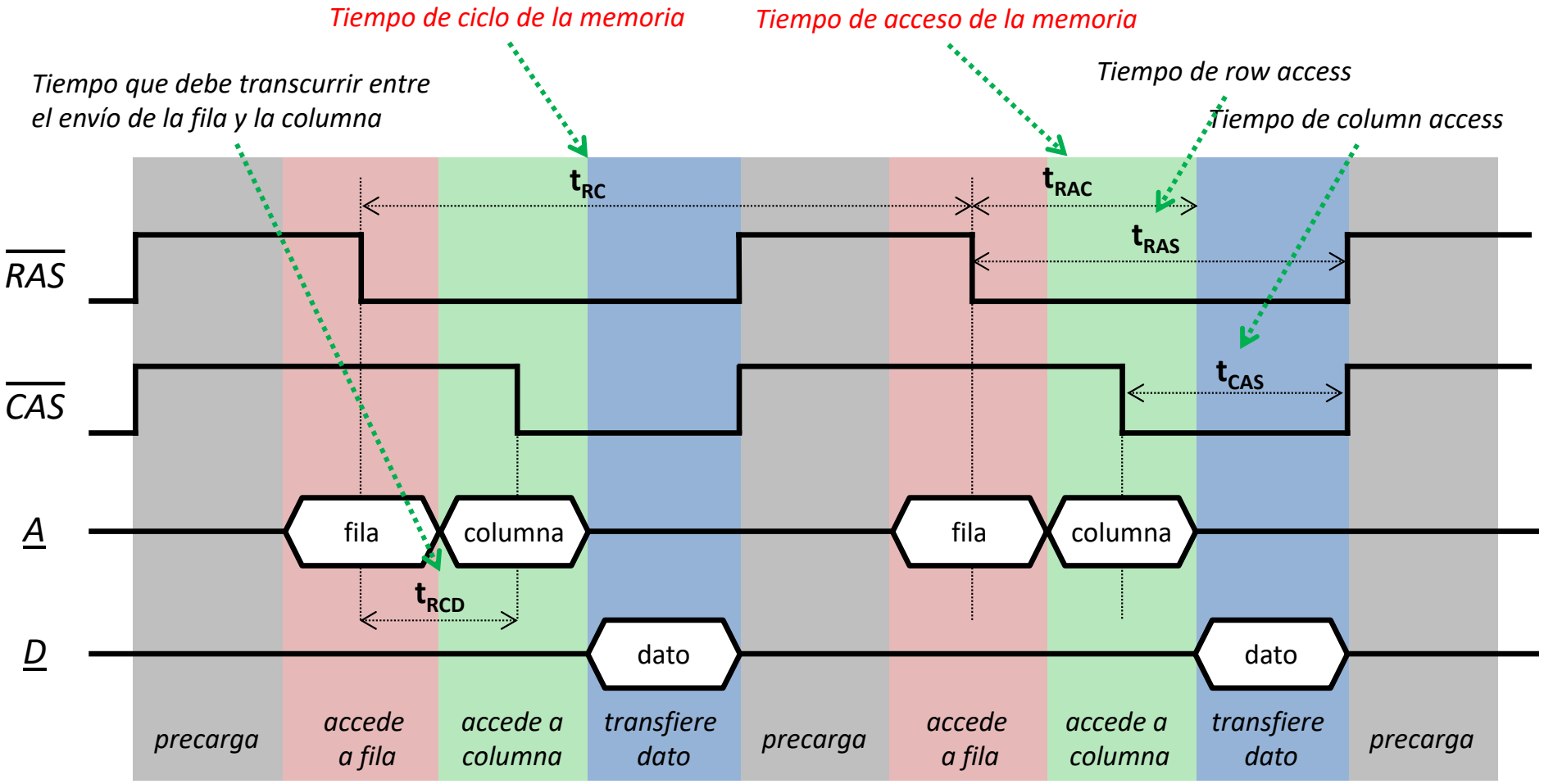
- En general, la organización física de las celdas de DRAM es independiente de la organización lógica de los datos.
  - Las DRAM suelen tener interfaces estrechos de datos k = (1b, 4b, 8b ó 16b)



Aspectos tecnológicos

Ciclo de acceso DRAM asíncrona (1970)

- DRAM (asíncrona): el acceso a cualquier dato siempre requiere indicar tanto su fila como su columna.

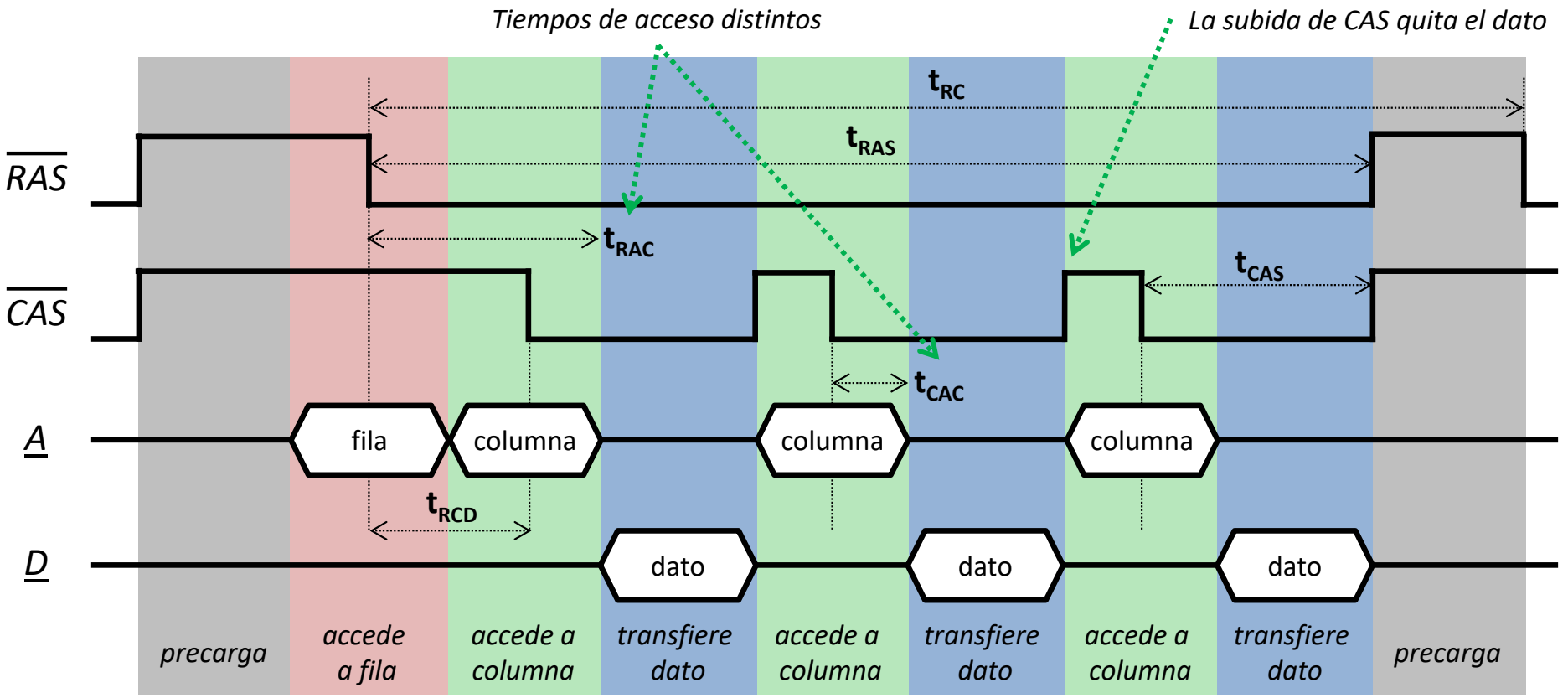


Ciclo de acceso simplificado

Aspectos tecnológicos

Ciclo de acceso FPM DRAM (1986)

- FPM (*fast page mode*) DRAM: el acceso a datos de una misma fila requiere indicar únicamente la columna.
  - Solo en el primer acceso a la fila se indica fila y columna.
  - Este primer acceso es lento, los sucesivos más rápidos.

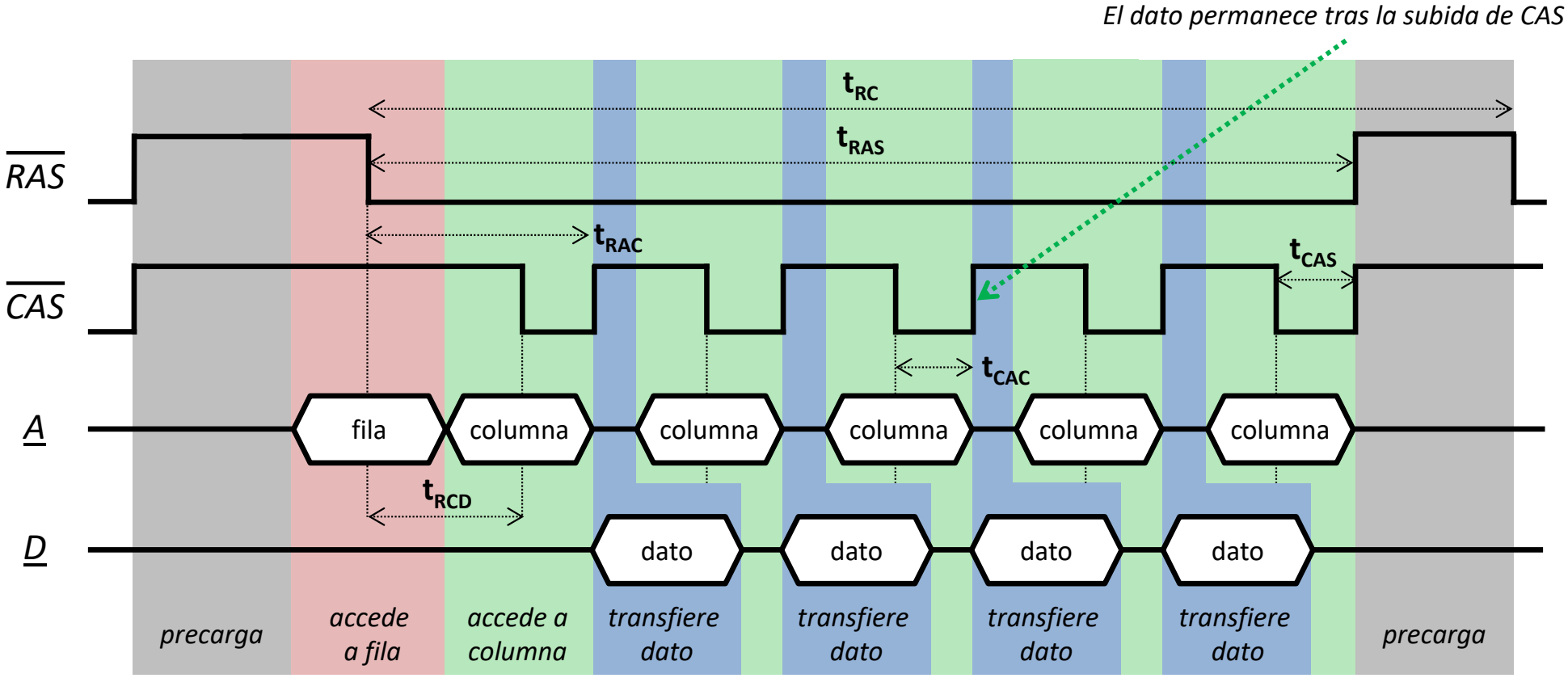


Ciclo de acceso simplificado

Aspectos tecnológicos

Ciclo de acceso EDO DRAM (1995)

- EDO (*Extended Data Output*) DRAM: permite indicar una nueva columna durante la transferencia de la anterior.
  - Acelera el acceso a datos de la misma fila.

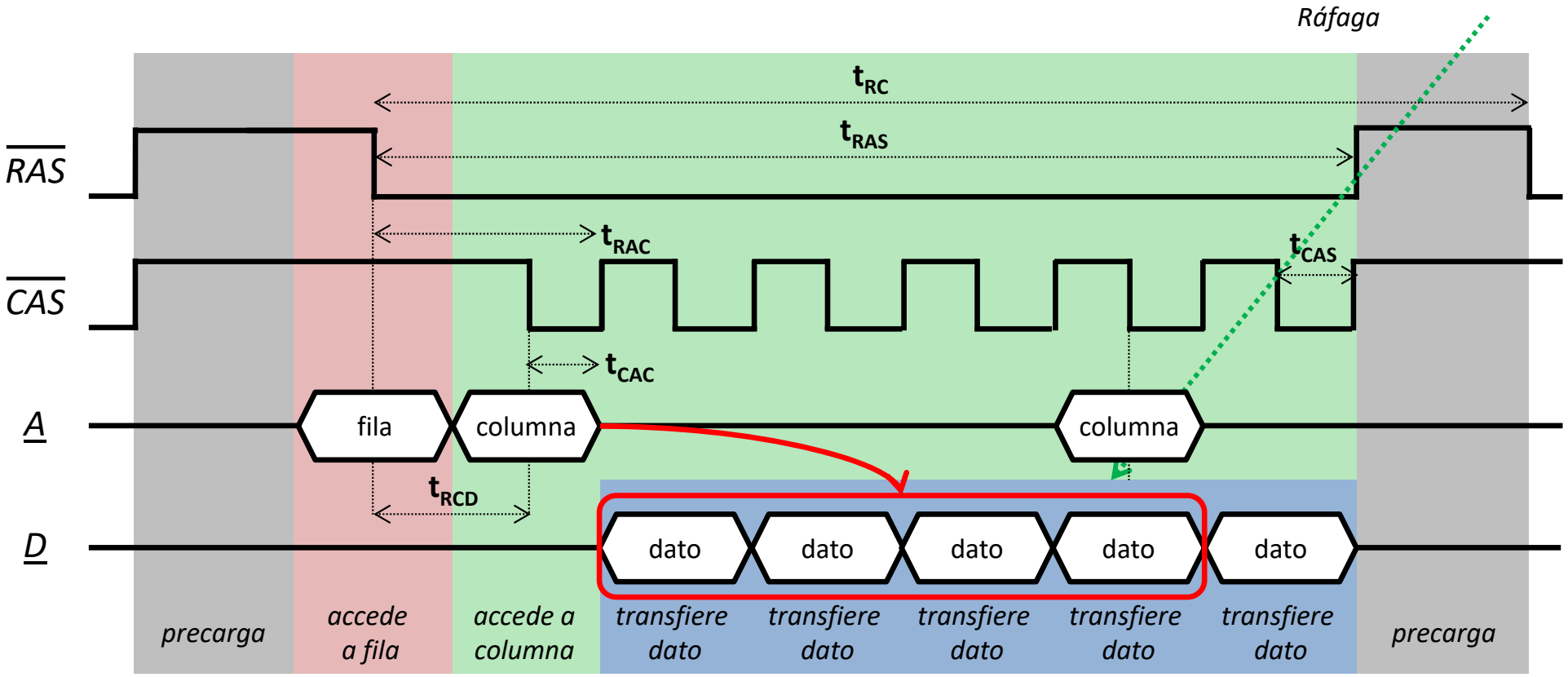


Ciclo de acceso simplificado

Aspectos tecnológicos

Ciclo de acceso BEDO DRAM (1997)

- BEDO (Burst EDO) DRAM: una vez indicada la columna, se accede a las columnas siguientes en ráfaga (*burst*).
  - Tiene un contador de columna interno que se actualiza en cada acceso.
  - El envío de una nueva columna inicia la ráfaga y finaliza la anterior.

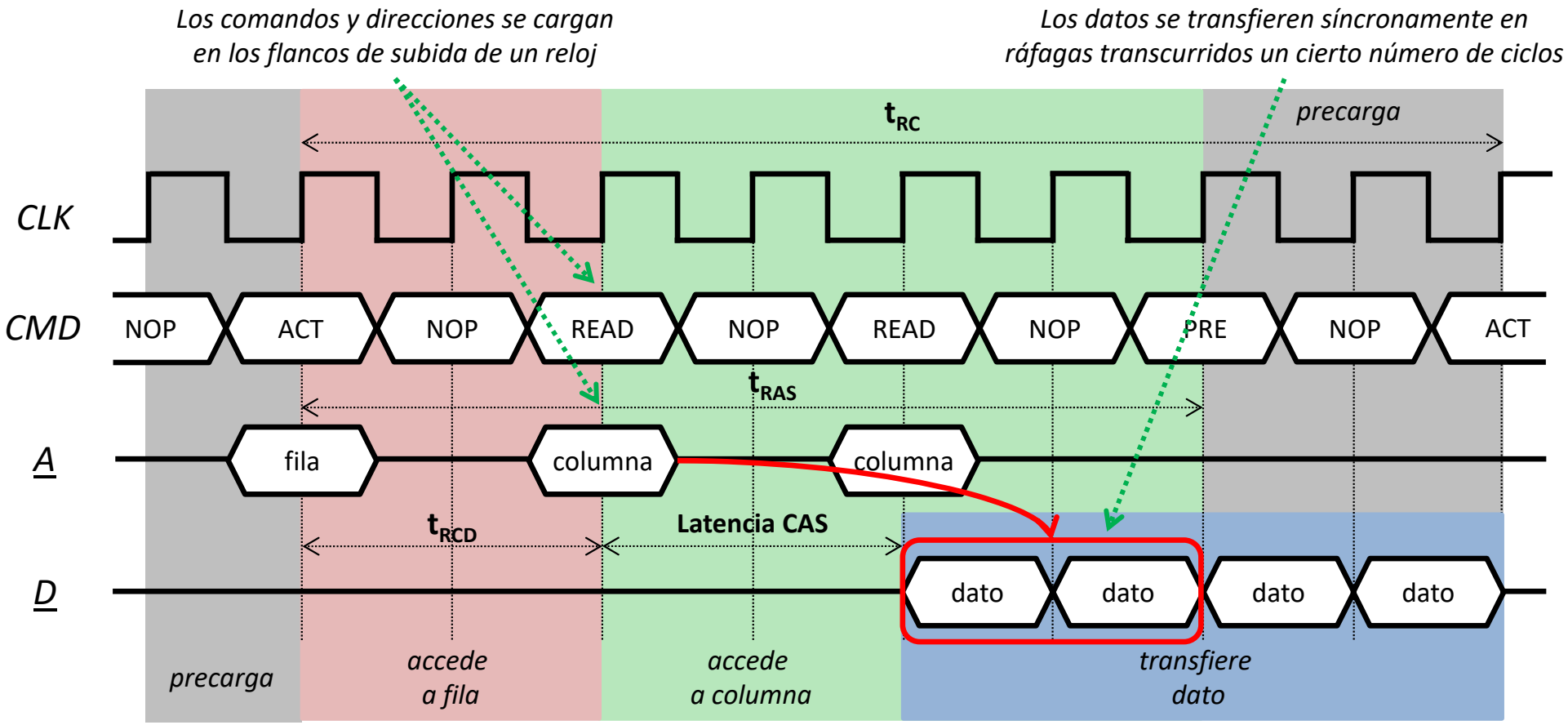


Ciclo de acceso simplificado

Aspectos tecnológicos

Ciclo de acceso SDRAM (SDR, 1997)

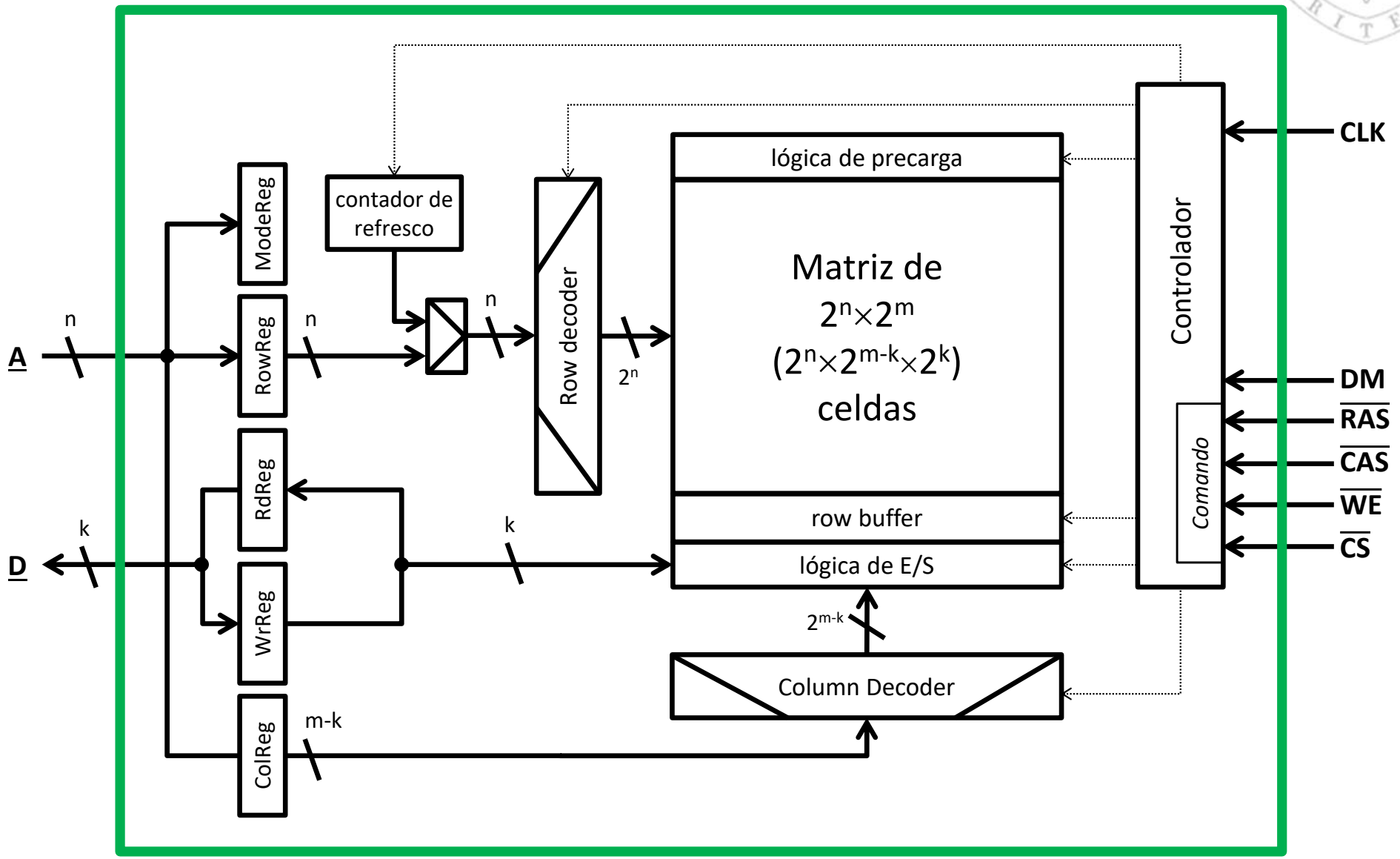
- SDRAM (Synchronous DRAM): memoria DRAM con interfaz síncrono.
  - Desacopla el interfaz de la compleja generación de pulsos asíncronos.
  - Se controla por comandos codificados en las líneas: CS, RAS, CAS, WE...



Ciclo de acceso simplificado

Aspectos tecnológicos

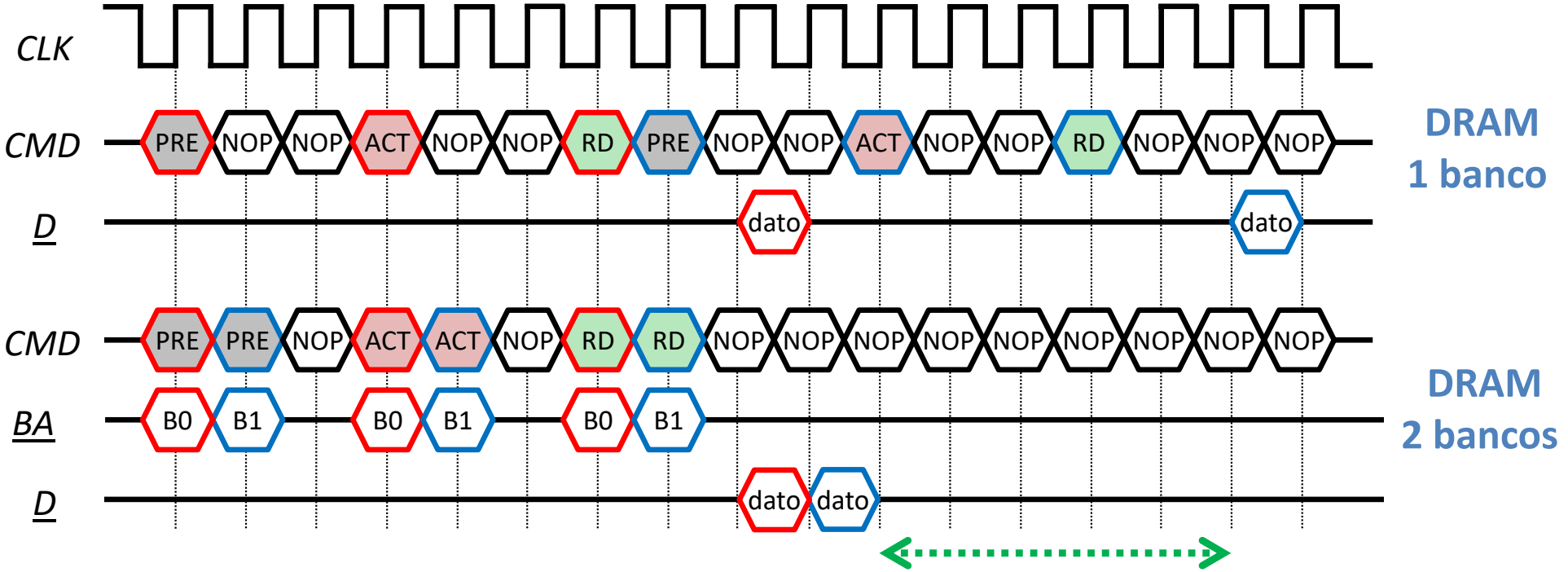
Memoria SDRAM (SDR): organización



Aspectos tecnológicos

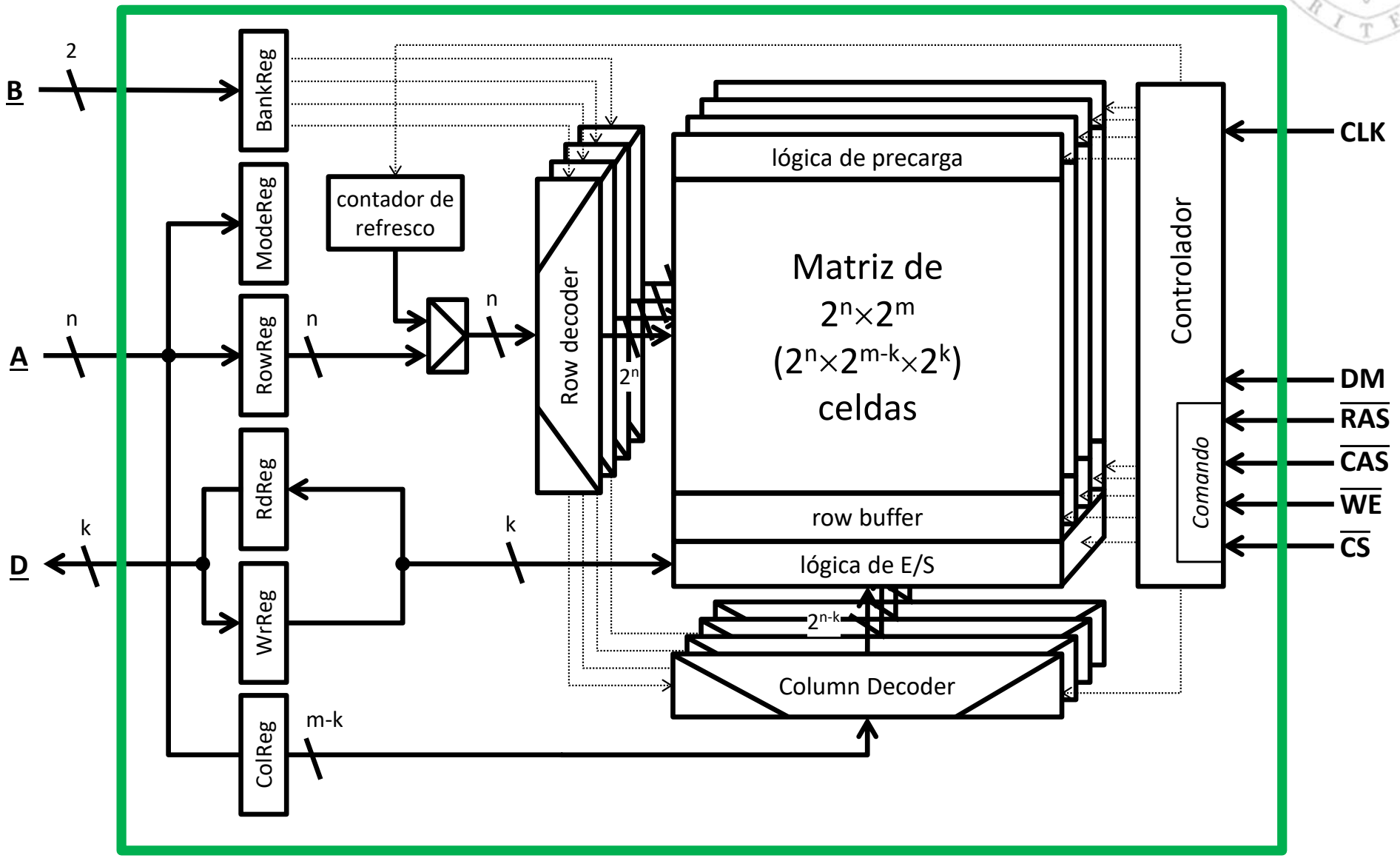
Memorias SDRAM (SDR) multibanco

- Para ocultar la latencia de acceso a los datos las SDRAM se organizan internamente en bancos accesibles en paralelo.
  - Un banco es una matriz de celdas distinta que opera independientemente.
  - Los bancos comparten pines de comando, dirección y de datos.
  - Añade una línea para indicar el banco a quien van dirigidos los comandos.



Aspectos tecnológicos

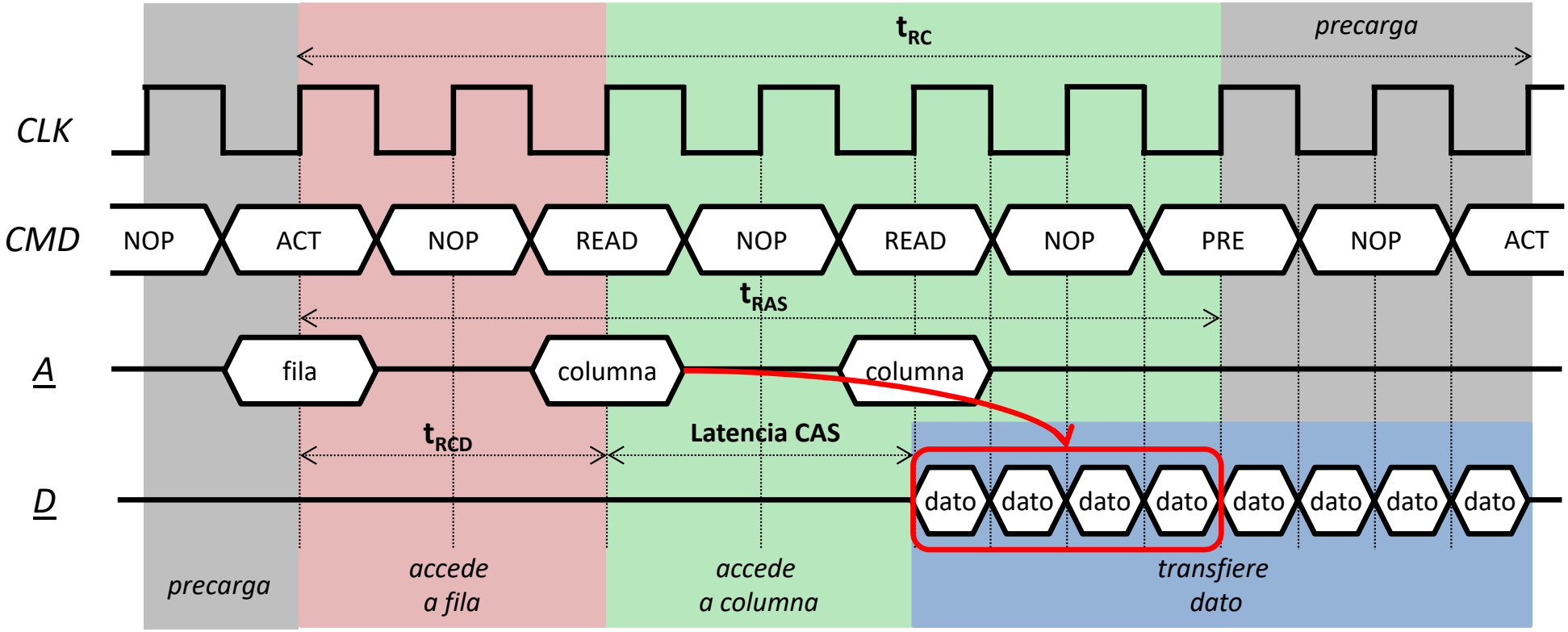
Memoria SDRAM (SDR) de 4 bancos: organización



Aspectos tecnológicos

Ciclo de acceso DDR (DDR1, 1998)

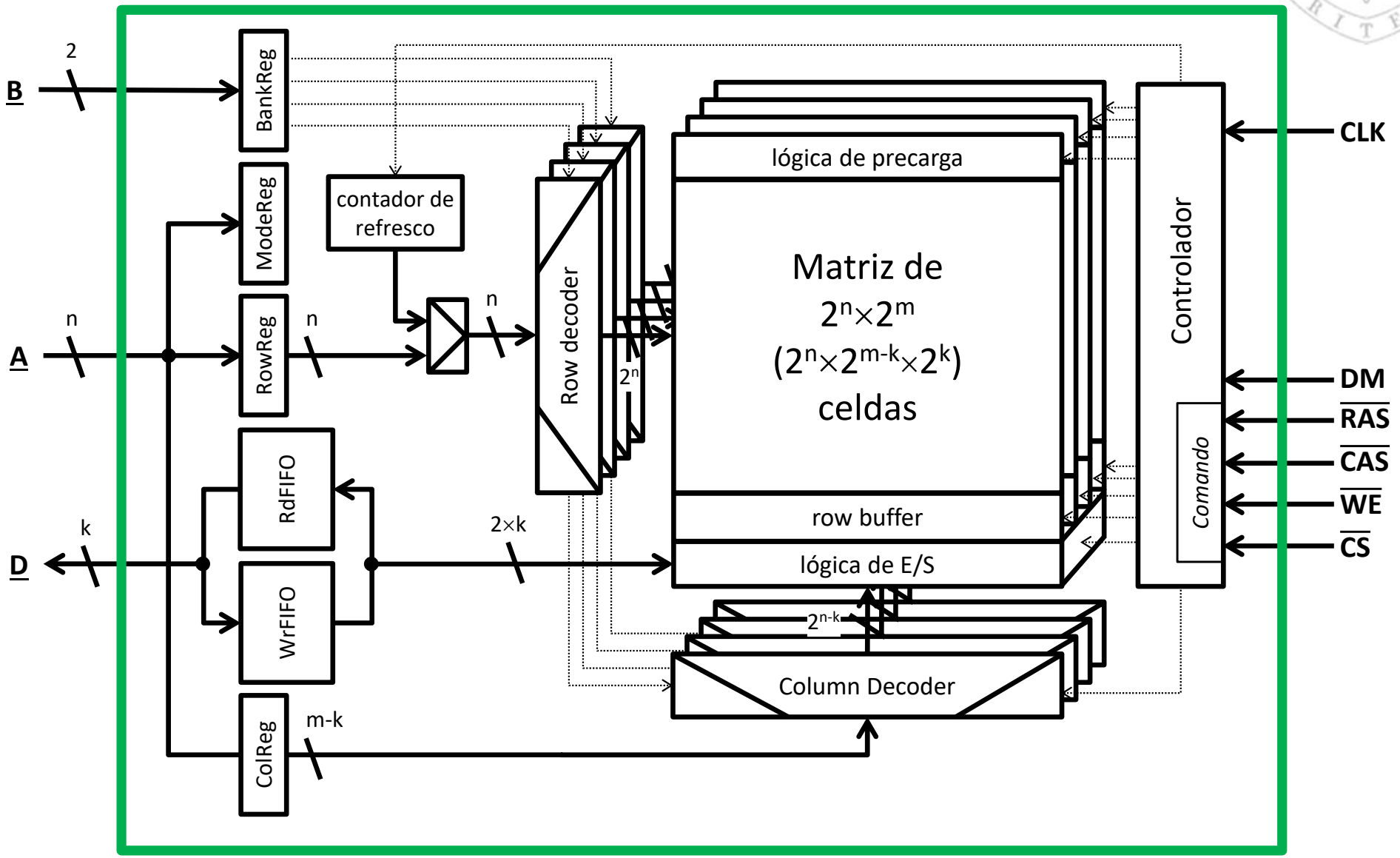
- DDR (*Double Data Rate*): memoria SDRAM que aprovecha ambos flacos de reloj para transmitir datos.
  - Por cada acceso a columna, se captan (*prefetch*) 2 datos del *Row Buffer*.
  - Duplica la tasa de transferencia sin aumentar la frecuencia de reloj.



Ciclo de acceso simplificado

Aspectos tecnológicos

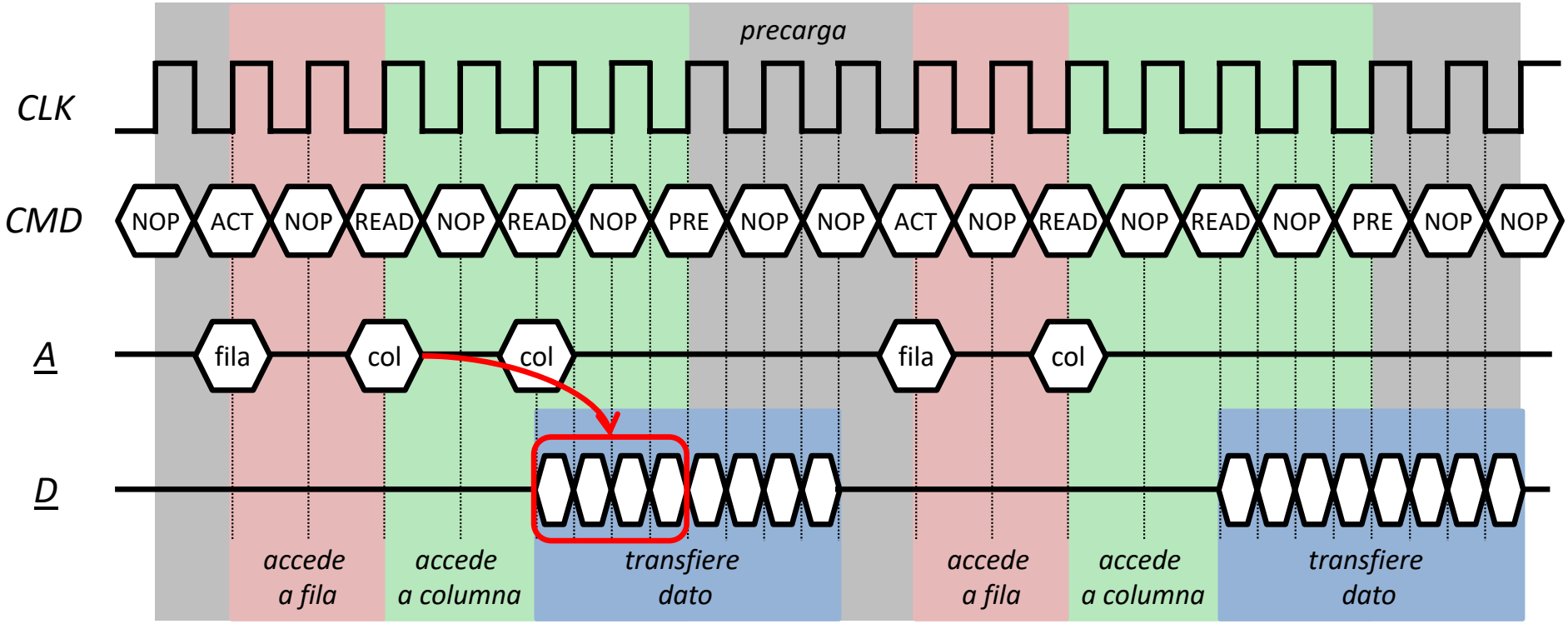
Memoria DDR: organización



Aspectos tecnológicos

Ciclo de acceso DDR2 (2004)

- DDR2 (*Double Data Rate 2*): duplica la frecuencia del interfaz respecto a la DDR manteniendo la velocidad de acceso a la matriz de celdas.
  - Por cada acceso a columna, se captan (*prefetch*) 4 datos del *Row Buffer*.
  - El interfaz alcanza frecuencias de reloj de hasta 400 MHz.

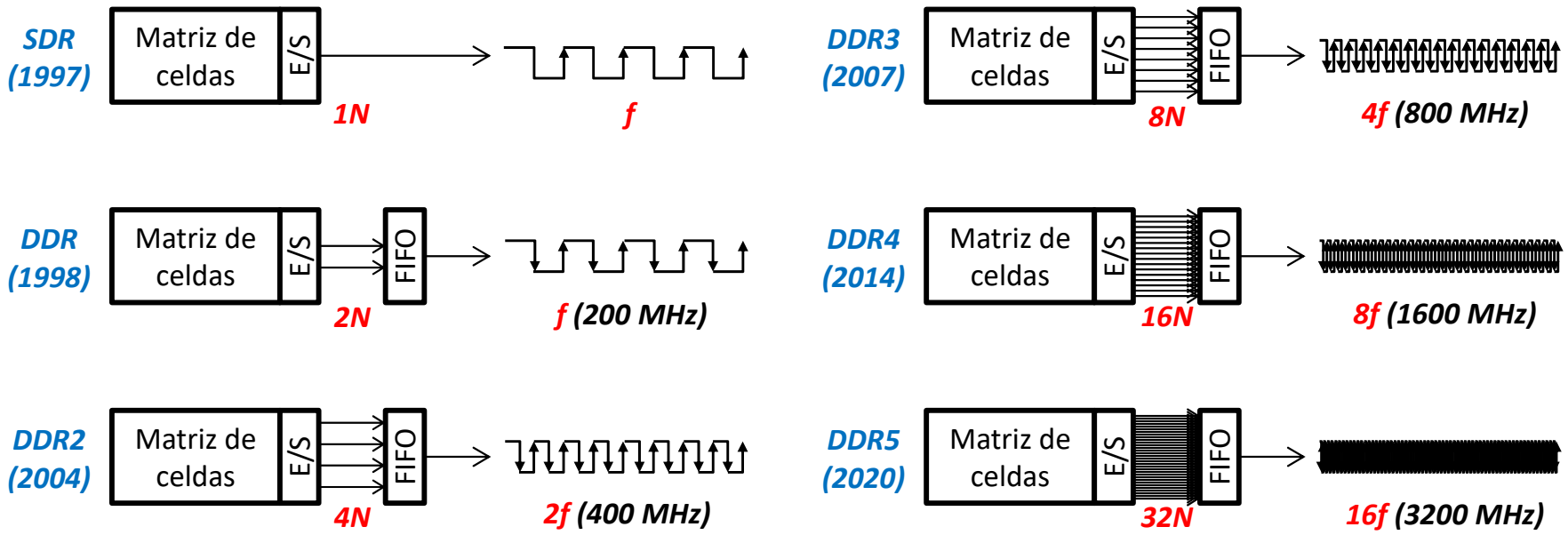


Ciclo de acceso simplificado

Aspectos tecnológicos

Generaciones de memorias DDR

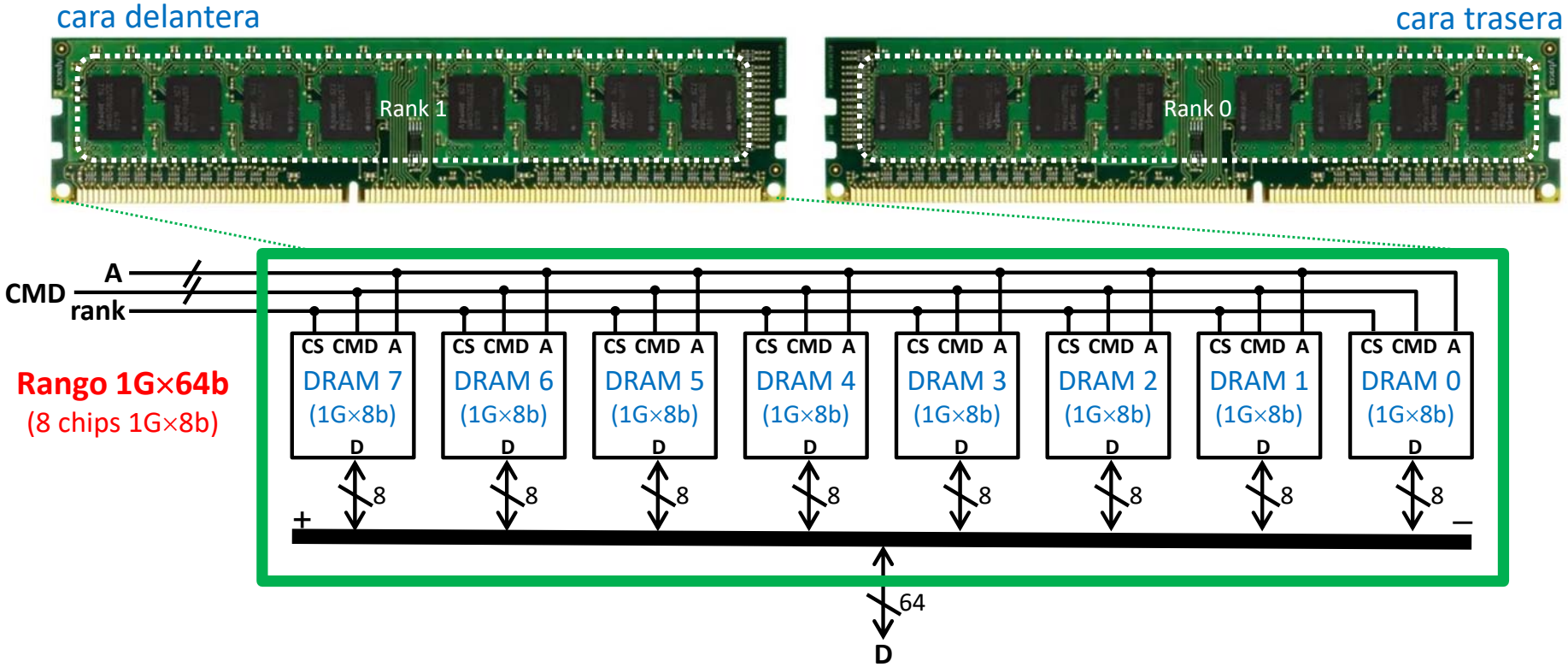
- En las sucesivas generaciones de DDR el tiempo de acceso a la matriz de celdas ha permanecido prácticamente constante.
- Sin embargo la tasa de transferencia ha crecido exponencialmente:
  - Cada generación duplica la frecuencia del interfaz.
  - Duplicando la cantidad de datos captados del *Row Buffer* por cada acceso.



Aspectos tecnológicos

Módulos de memoria SDRAM

- Los interfaces de datos de los chips de SDRAM son estrechos (4/8/16b)
  - Se organizan en rangos (*ranks*) para formar un interfaces más anchos
    - Se activan a la vez, comparten dirección y comando, los datos se unen en paralelo.
  - Los rangos se agrupan en módulos DIMM (*Dual Inline Memory Module*)
    - Comparten dirección, comando y datos pero se seleccionan independientemente.



Aspectos tecnológicos

Controlador de memoria

- El interfaz de las memorias SDRAM es tan complejo que la CPU/MC se conecta a ella a través de un controlador de memoria que:
  - Recibe y sirve peticiones de acceso por parte de la CPU/MC.
  - Transforma estas peticiones en secuencias de comandos.
  - Almacena los comandos y reordena su envío para mejorar el rendimiento.
    - Llevando pista del estado de cada rango, banco, fila, etc...
    - Intercalando accesos con los ciclos de refresco.
    - Respetando las múltiples (+50) ligaduras físicas de tiempo existentes.
  - Gestiona errores de lectura.
  - Supervisa el consumo y temperatura de la SDRAM.

Aspectos tecnológicos

Memorias SRAM vs DRAM (DDRx)

Característica	SRAM	DRAM (DDRx)
Transistores/bit	6T/b (2 terminales)	1T/b (1 terminal)
Tiempo de acceso a celda	Muy bajo y fijo	Alto y variable
Densidad de integración	Baja	Muy alta
Ubicación	On-chip	Off-chip
Coste/bit	Alto	Bajo
Lectura	No destructiva	Destructiva
Necesidad de refresco	No	Si
Organización de celdas	2D	3D (multibanco)
Dirección multiplexada	No	Si
Interfaz	Simple y asíncrono	Complejo y síncrono

Acerca de *Creative Commons*

- Licencia CC (Creative Commons)
  - Ofrece algunos derechos a terceras personas bajo ciertas condiciones. Este documento tiene establecidas las siguientes:

Reconocimiento (*Attribution*):
En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.

No comercial (*Non commercial*):
La explotación de la obra queda limitada a usos no comerciales.

Compartir igual (*Share alike*):
La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Más información: https://creativecommons.org/licenses/by-nc-sa/4.0/